1985

A cell level control structure for a flexible manufacturing assembly cell /

Joseph Allen Wetzel
Lehigh University

A CELL LEVEL CONTROL STRUCTURE FOR A FLEXIBLE MANUFACTURING ASSEMBLY CELL

by

Joseph Allen Wetzel

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Manufacturing Systems Engineering

Lehigh University

1984

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

12/12/84
(date)

Professor in Charge

Director, MSE Program

Chairman of Department

ACKNOWLEDGMENT

I would like to thank my thesis advisor, Dr. Richard Roberts, for his time and effort as he guided me in the preparation of this document. And to my wife, who endured many lonely nights and untold miseries in support of my efforts, I say thank you for all your love and support.

TABLE OF CONTENTS

LIST OF FIGURES

1. ABSTRACT

Flexible manufacturing cells for assembly are beginning to be implemented in industry. These assembly cells bring together programmable robots, material transfer devices, and other special purpose equipment for the purpose of assembling a group of related parts, often called a part family. The interactions that result from bringing together multiple devices and systems to form an assembly cell, require coordination and control. As a result, control systems are developed to manage the cell through the coordination and integration of the cell's components.

This thesis begins with an examination of cellular manufacturing in which cellular manufacturing is defined relative to assembly cells and cell control. The nature of assembly is investigated in order to identify the assembly issues which are pertinent to flexible manufacturing and cell control. In an effort to gain a deeper insight into assembly cells, a detailed study is made of an assembly cell currently being implemented in a manufacturing firm. The hierarchical control theory being developed by the Industrial Systems Division at the

National Bureau of Standards is discussed, and some basic principles of hierarchical control are used as the foundation for an assembly cell control structure. Then the concepts of hierarchical control, flexible manufacturing, and assembly, are used to develop a layered control structure for assembly cells. This control structure provides the logical framework necessary for the development of assembly cell control systems which are easily maintained and modified.

2. INTRODUCTION

New more competitive markets are forcing modern manufactures to operate within a set of constraints that seem to keep tightening. Productivity growth has slowed, labor rates have increased, customers are demanding higher quality products and shorter lead times, interest rates have risen into a new operating range, and the cost of carrying inventory is becoming prohibitive. As a result, todays manufacturing firms are looking for innovative new solutions to a set of problems which are no longer tolerable. These solutions will require the development of new manufacturing systems which will be capable of achieving; significant productivity increases, improvements in quality, improvements in the utilization of capital, and meaningful reductions in inventory.

Coincident with the changing economic environment has been the development of programmable automation. Programmable automation has been developed to provide the flexibility and adaptability required in today's manufacturing environment. Programmable automation is especially well suited for batch manufacturing where the high cost of special-purpose automated equipment can not be justified. Some forms of programmable automation such as NC (numerically controlled) machining have been in use for some time and are becoming mature technologies. However, other forms of programmable automation are just beginning to be applied to manufacturing in meaningful numbers. Robotics is one of these technologies. As technological improvements continue to be made in the field of robotics, these tools are being applied in increasing numbers. Although today the majority of robot implementations fall within a few application areas like simple material handling, welding, machine tending, and painting, there is a move to begin to apply them in other areas, such as assembly. It is becoming clear that programmable devices, like robots, will be the key ingredients in new manufacturing systems.

The concept of cellular manufacturing is being used to bring together an assortment of equipment for the purpose of manufacturing a group of related parts, often called a part family. Cellular manufacturing is based on concepts founded in what is now called group technology. Automated assembly cells arise out of the need to improve the productivity of batch assembly. Typically, assembly cells contain programmable robots, material handling devices, and other special purpose equipment.

The interactions that result from bringing together multiple devices and systems to form an assembly cell, require coordination and control. As a result, some sort of control system is developed to provide the coordination and integration necessary to transform a collection of loosely related equipment into a smoothly operating integrated assembly cell. Cell control systems for automated assembly cells and the control system structure are the major topics addressed in this thesis.

This thesis will begin with an examination of cellular manufacturing. Then the nature of assembly is investigated in order to identify the assembly issues pertinent to cell control. This is followed by a detailed study of an assembly cell which is in the process of being implemented by a small parts manufacturing firm. This cell is used to illustrate one type of assembly cell being implemented in industry today, and provides another opportunity to identify important control issues. Then the concepts developed in previous sections are used to develop a multi-layer cell control structure. This paper concludes with several recommendations for further study.

3. THESIS OBJECTIVES

The objective of this thesis is to develop a control architecture for assembly cells. The cell control system structure should support the development of control systems which are easily maintained and modified. The control structure should be capable of growing with the cell as new capabilities and new devices are included in the cell, and should be designed so as to facilitate its integration as part of larger CIM systems.

It is important to note that this cell control architecture is a logical structure. It is intended to provide an approach to the design of a control system that reduces complexity and results in systems that are flexible and adaptable. It is not the objective of this work to define implementation, and issues such as the selection of appropriate hardware, will not be discussed. The very power of a such a logical control structure is that while providing a structured approach to cell control system development, system developers are free to implement control in the way that best meets the needs of the cell. Issues involving implementation, such as the hardware/software boundary, will change with time as hardware capabilities expand and new software approaches and programming languages become available. Thus, the intent is to develop a logical control structure which is useful, yet is able to accommodate several generations of implementation.

4. CELLULAR MANUFACTURING

Cellular manufacturing systems have emerged as an alternative to the more traditional forms of manufacturing. Cellular manufacturing combines processes, people, and machines in order to manufacture a specific group of parts, often called a part family.[1] A manufacturing cell is defined as a cluster or collection of machines designed and arranged to produce a specific group of parts. A flexible manufacturing system (FMS) can then be defined as a highly automated cellular manufacturing system. Some of the benefits of cellular manufacturing systems are: [2,3]

* increased productivity
* improved quality
* reduced inventories
* reduced work in process
* reduced manufacturing lead times
* increased utilization of capital
* improved management control
* improved quality of work life for employees

Much of the early work in cellular manufacturing was, and continues to be, directed toward machining cells. Typical machining cells contain one or more NC-machine tools which are serviced by robots. The robots perform such tasks as tool changing and part loading and unloading.

Kusiak has developed an FMS classification scheme which is oriented toward machine tool systems.[4] He divides an FMS into five classes based on the number of numerically controlled (NC) stations and their arrangement:

1) flexible manufacturing module
2) flexible manufacturing cell
3) flexible manufacturing group
4) flexible production system
5) flexible production line

A flexible manufacturing module (FMM) is defined by Kusiak as an NC-machine tool which is further automated by a tool loader, a pallet loader, or both. A flexible manufacturing cell (FMC) consists of multiple NC-machine tools which are enhanced with automatic loading capabilities. In both an FMM and an FMC the automatic loading functions are often performed by a robot. Kusiak continues by defining a flexible manufacturing group (FMG) as a collection of FMM's and FMC's which are joined by a material handling system. The FMG also operates under computer control. A flexible production system (FPS) consists of FMM's whose functions vary among machining, assembly, and fabrication. Kusiak concludes by defining a flexible manufacturing line (FML) as a set of NC-machine tools joined by a material handling system.

4.1. GROUP TECHNOLOGY AND CELLULAR MANUFACTURING

A total systems approach must be used in the design and development of manufacturing cells to insure the flexibility required by an increasingly dynamic environment. An important aide in the development of flexible manufacturing systems is group technology. Group technology is a systems based rationale which identifies and exploits the sameness of the items and processes used in manufacturing.[1]

One of the objectives in applying group technology is to organize parts and components into groups or families. A classification and coding system is used to formulate the part families. Before part families can be formed a part database must be established. This database may contain information such as; part number, geometry, material, function, process equipment, machine utilization, and operating procedures.[4,5]

Once the data has been collected a coding scheme, based upon the characteristics with which parts are classified, is used to formulate a code for each part or component. Coding systems can be tailored to meet the objectives of the particular firm. For example, firm A may want its coding system to aid in the reduction of design replications. This coding system would emphasize geometry and functionality. Now suppose that firm B wants its coding system to aid in the development of production plans and routings, its coding system will place emphasis on the production processes for each part.

Once a database of coded parts exists, the data contained within each part's group technology code can be used to build families of parts. Once part families have been created in a systematic manner, the process of developing a manufacturing cell has been simplified. All of the information used in the classification process and contained within the part code, forms a set of specifications for the manufacturing cell being developed for a particular family or families. As a result, group technology provides the opportunity to analyze geometry, process, and material information for a family of parts during the design and development of a flexible manufacturing cell. In essence group technology concepts can be used to provide a systematic approach toward the development of part families for which flexible manufacturing systems can be developed.

4.2. ATTRIBUTES OF MANUFACTURING CELLS

Manual operations provide flexibility without consistency and speed. Hard automation provides speed and consistency without flexibility.[3] Cellular manufacturing is a manufacturing philosophy aimed at providing consistency, flexibility, and speed in what can be called a flexible manufacturing system (FMS). A total systems approach to cellular manufacturing will provide the means to: increase productivity, improve quality, reduce inventories and work in process, reduce manufacturing lead times, increase utilization of capital, improve management control, and improve the quality of work life for employees.

The flexible in flexible manufacturing systems means that the cell must be capable of manufacturing a class or family of parts with similar geometry, processes, and material characteristics. It means that part change-overs are simple and setup time is minimal. In order to reduce inventory and work-in-process, batch sizes have to be reduced. As batch sizes are reduced part change-overs become more frequent and setup time is distributed across fewer parts. It quickly becomes evident that in order to economically reduce batch sizes set up time must be reduced and if possible eliminated. This also means that new approaches to tooling and fixtures must be developed. Tooling must be easy to change and when ever possible it must be designed so as to be suitable for a complete part family, thereby relieving the need for a tool change. As the adage goes, "reduce, simplify, and eliminate". We would do well to re-learn this lesson that we taught so well to the Japanese.

Flexible also means that the devices which make up the cell are intelligent and programmable. Programmable devices such as NC-Machine tools and robots can be instructed to process a different part in a part family by providing the device with a new program. Flexibility can be extended even further by adding sensory and communication capabilities to the devices. With sensory and communication capabilities the device is able to gather more information about its environment and better control it process. Communication capabilities also permit controlled interaction between various devices within the cell as well as providing for control at the cell level.

4.3. CELLULAR MANUFACTURING DEFINED

For the purpose of this work it was necessary to develop a definition for cellular manufacturing which is not restricted only to machining but which is general enough to include fabrication and assembly.

Groover's definition of a flexible manufacturing system (FMS) is used to define cellular manufacturing. According to Groover, an FMS consists of a group of processing stations which operates as an integrated system under computer control.[2]

The concept of cellular manufacturing is closely coupled to group technology. Cellular manufacturing combines people, processes, and machines to form an FMS to manufacture a specific group of parts, often called a part family. A manufacturing cell of this type is characterized by the use of intelligent programmable devices, the use of computer control at the station/device level as well as at the cell level, the use of sensors, the interconnection of devices for communication, and by a cell level interface to human operators and other factory systems.

5. ASSEMBLY CELLS

Today automated assembly is most frequently associated with hard automation and serial assembly lines, such as those which are found in the automotive industry, appliance industry, and other industries which manufacture high volumes of assembled parts. These systems have evolved out of the concepts of standardized parts and serial moving assembly lines introduced by Eli Whitney and Henry Ford. As new technology was introduced it was possible to begin to automate the simpler assembly tasks and reduce the direct labor required for assembly. With the advent of pneumatic and electronic logic it became possible to automate more complex assembly tasks and hard automation became even more common place.[6]

Generally, this kind of assembly automation is only suitable for high volume products that exhibit relatively little change during the product life cycle. The significant investment in equipment, system development time, and the lack of flexibility associated with hard automation assembly, make it inappropriate for batch manufacturing. Where parts are manufactured in batches and part changeovers are frequent, assembly is primarily a manual operation. According to Westinghouse statistics, batch manufacturing accounts for 75% of all manufacturing and 22% percent of the U.S. gross national product.[7] At the same time it has been estimated that assembly operations account for the largest portion, 22%, of the total labor costs for all durable goods.[8] Surely batch manufacturing offers many opportunities for increased productivity through new forms of flexible automation.

With the development of advanced programmable robots and inexpensive microcomputers comes the potential for flexible automated assembly. Todays advanced robotic and computer technology, when coupled with a total systems approach to control, permit the development of automated assembly systems which are flexible enough to economically assemble small batches of parts. These assembly systems are being developed to assemble groups of similar parts called part families. A recent survey shows that robots are expected to play a major role in assembly in the coming years. In 1980 six percent of some 5,000 robots at work in the U.S. were used in assembly, and projections indicate 37% of an estimated 100,000 robots will be used in assembly by 1990.[9]

5.1. TYPES OF ASSEMBLY

Assembly is the fitting together of component parts. This includes mating and joining. Parts may be mechanically fastened through the use of screws, nuts, rivets, and other assembly operations such as press fitting, swaging, and staking. Parts may also be joined through processes such as welding, brazing, soldering, and adhesive bonding.

Automatic assembly is a term that refers to the use of mechanized and automated devices to replace manual assembly operations. Hard automation in serial assembly lines is just one type of automated assembly. Another form of automated assembly deals with the assembly of parts consisting of a relatively small number of components. Assembly of these parts is manageable within the confines a of single assembly machine. Groover takes this perspective in his discussion of automatic assembly machines. Groover explains that automated assembly machines typically consist of a transfer system, automatic workstations, and a manual assembly station. The automatic workstations typically consist of component parts storage, feeding, and orienting mechanisms. The assembly tasks which are too complex or too expensive to mechanize are performed by humans at the manual station.[2]

Here again automatic assembly machines exhibit many of the same problems associated with large serial hard automation assembly lines. Automatic assembly machines lack the flexibility required to assemble batch volume products which are also characterized by frequent design changes. These machines also represent a significant capital investment which often renders them unsuitable for products with a short life cycle.

The integration of advanced programmable robots, computer control, sophisticated material handling systems, complex sensory systems, and fixed automation, can yield assembly systems which are both flexible and adaptable. These adaptable programmable assembly systems will be capable of assembling whole part families as well as being able to be adapted to new product designs. With 22% of the total labor cost for all durable goods attributed to assembly, these systems have the potential for large payoffs.

5.2. ASSEMBLY CELL COMPONENTS

In chapter four cellular manufacturing was defined as a system combining people, processes, and machines for the purpose of producing a specific group of related parts. Such systems were said to be characterized by the use of intelligent programmable devices, the use of computer control, the use of sensors, the interconnection of devices for communication, and by a cell level interface to human operators and other factory systems. Such systems whose purposes are to assemble a group of related parts are called assembly cells. Typical assembly cells consist of robots, material handling systems, parts feeding and orienting systems, computers, cell and device control systems, and special purpose work stations.

5.2.1. ROBOTS

Certainly there is a whole spectrum of robots, from simple pick and place robots to complex six axes robots with sophisticated grippers. Generally, more degrees of freedom mean more capability, therefore those robots at the high end of the spectrum contribute most to the flexibility of an assembly cell. Work being done at the University of Massachusetts has indicated that a two arm, four degrees of freedom robot is a basic requirement for assembly.[11] As these new robot designs emerge and are incorporated into automated assembly systems, it will become feasible to automate more and more manual assembly tasks.

A robot's effectiveness is often limited by the end of arm tooling or gripper. For this reason part handling tools (grippers) are being researched in the hopes of designing more versatile grippers capable of serving a variety of assembly needs. Three finger grippers, four finger grippers, and grippers with reconfigurable fingers are all being investigated.[11] According to Mutter, compliant mountings such as RCC (Remote Compliant Centers) devices are a must when a part or parts have to be inserted or fitted into another. These devices help to correct for misplacements and misalignment.[12]

Advanced programmable robots provide much of the flexibility and adaptability in an automated assembly cell. Much of this flexibility is achieved because a robot can be directed to carry out different tasks by simply providing the robot controller with different programs. To a large extent, robot flexibility can be measured by how easily the robot is programmed. Off-line programming and robot program simulation systems are adding to the flexibility of robots. Just as the physical and mechanical properties of robots are continually being improved, so are robot programming languages.[10] Advances in high level programming languages for robots will continue to increase the flexibility and inherent adaptability of assembly systems containing robots.

At times there is a tendency to view the robot and the robot controller as two separate devices. This is a result of the advances in robot controllers that enable not only high level robot programming but additional ancillary device communication and control. This has lead to the practice of using the robot controller to control other devices in the manufacturing cell. Although this may be a satisfactory solution in a few cases it should not become a standard control strategy. This will be discussed at greater length later.

5.2.2. FEEDERS / ORIENTERS

Mechanized assembly processes generally require that component parts are presented to an assembly workstation in the proper orientation. The variety of different component geometries is often a problem in the design of feeding/orienting mechanisms for automated assembly systems. Often new feeding/orienting solutions must be developed for each component part. Many times parts feeders must deliver parts at high rates of speed and with high reliability.[2,12] If a part feeder goes down, the system will be starved of the component part and will be forced to either produce incomplete assemblies or stop operation.

Parts feeding and orientation has a major impact on the flexibility of assembly cells. Conventional parts feeding and orienting mechanisms, such as vibratory bowl feeders, represent significant investments and offer only limited flexibility. Additionally, it is common to use specially designed jigs and fixtures to precisely locate component parts. Many times the investment in these fixtures is significant and design changes to the part being fed and located mean sometimes costly modification or replacement of such tooling, thereby limiting flexibility.

If truly flexible assembly systems are to be implemented, flexible parts feeding and orienting systems capable of feeding multiple parts must be developed. Generic tooling which is suitable for whole part families and which can be easily reconfigured or modified must also be developed. The tooling issue can be regarded as more of an attitude toward the design. Clearly, the flexibility of assembly systems can be improved today, without significant technical innovations, if design philosophies are adopted which emphasize flexibility. Group technology concepts provide much of the foundation for such design philosophies. Surely it is much easier to design flexibility into tooling by exploiting the similarities found in a well defined group of parts which have been identified and collected using group technology techniques.

Much of the inflexibility of existing feeding and orienting systems can be removed through the use of sensors and programmable feeding systems. One approach is to feed parts without regard to orientation. Then a sensor system is used to sense the part's orientation. The part's orientation is then provided to the robot which can orient the part prior to the assembly operation. It is possible to design such systems so that simple orientations for a family of related parts are obtained by the feeding mechanism, and then a sensing system is used in conjunction with the robot to obtain the final orientation required for assembly. These systems offer extended flexibility in that the sensors can be easily reconfigured for different parts, and the robot can be provided a new program. The philosophy behind programmable feeders is, a low cost device capable of feeding a wide variety of parts into the correct orientation ready for pickup by a robot arm. Work under way at the University of Massachusetts is investigating modifications to a moving belt system originally designed by G. Boothroyd. This feeder system is based on two parallel moving belts, a set of mechanical orienters, and a sensor system. Both an active and a passive system are being investigated. In the active system a part's orientation is determined by sensors, and then mechanical orienting devices are manipulated to put the part into the required orientation. The passive system simply detects orientation and rejects those component parts which are improperly oriented.[11]

For flexible assembly systems for batch manufacturing to become a reality, there must be innovation in parts feeding and orientation. This innovation must lead to flexibility, adaptability, and the increased utilization of the capital applied to feeding and orienting.

5.2.3. SPECIAL PURPOSE WORKSTATIONS

It is not uncommon for assembly cells to include other operations which are normally performed just prior or just after assembly. Examples of such operations include metal forming, plastic molding, drilling and tapping, printing, and various inspection operations.[2] These workstations must be designed and implemented so that they are flexible and adaptable. If not, they will severely reduce the flexibility of entire assembly system.

5.2.4. COMPUTERS AND CONTROL SYSTEMS

Computers and control systems are used in assembly cells for device control and cell management. A robot is a complex device and control of the motion of a robot arm requires significant computational capability. For this reason various types of computers are being used as robot controllers. Computers are also being used to control a variety of other devices and systems such as automatic guided vehicles, material handling systems, and inspection systems. These computers and their control software are usually being used for low level machine or device control.

Other more complex computers and control software are being used to manage the assembly cell. These computers and control systems manage the integrated assembly cell as a system comprised of various devices and subsystems. The cell control system allocates resources, sequences cell activities, coordinates devices, provides an interface to the operator and other factory systems, and responds to errors. Cell control will be discussed at greater length later in this paper.

5.3. ASSEMBLY CELL CONFIGURATIONS

Cell configuration refers to the organization of cell components and their associated task assignments. To a large extent, the types of robots and their configuration within the assembly cell, determine the cell's capabilities. There are two basic cell configurations, serial and parallel.

5.3.1. SERIAL LINES

In serial cell configurations the assembly moves in a fixed sequence through a series of workstations each of which performs its assigned task, see figure 5-1. A workstation contains a robot or some other special purpose device. In electronics manufacturing typical serial systems transport the product from workstation to workstation by means of a conveyor and precision pallets. At each station the pallet is stopped, precisely located, and an operation is performed on the product.[6] The cycle time for serial lines is equal to the sum of the cycle time at the slowest station plus the transfer time between stations. Usually each workstation performs a single task or several simple tasks and proper line balancing is required to achieve high device utilization. Since each workstation performs a different task or tasks, all stations must be operational for the cell to be operating, unless work-in-process buffers are placed between workstations.

5.3.2. PARALLEL LINES

Parallel configurations consists of identical workstations. Each workstation within the cell performs all the operations assigned to the cell, see figure 5-2. Within the cell one product will pass through one workstation and the next product may pass through a

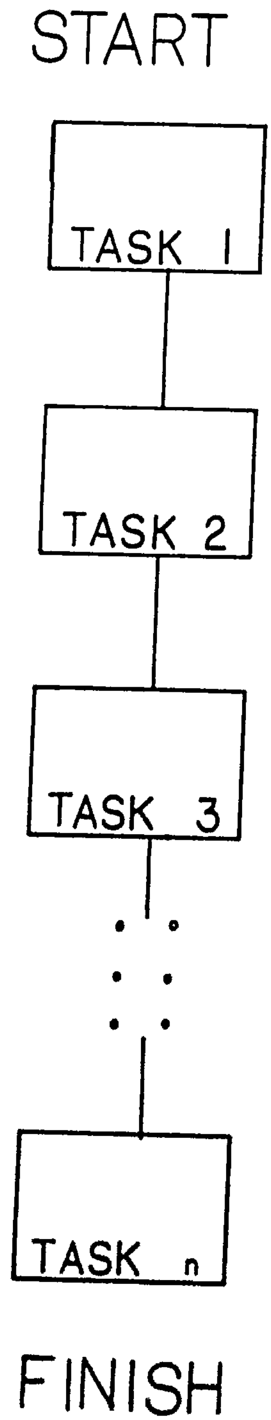


Fig. 3-1: Serial Configuration

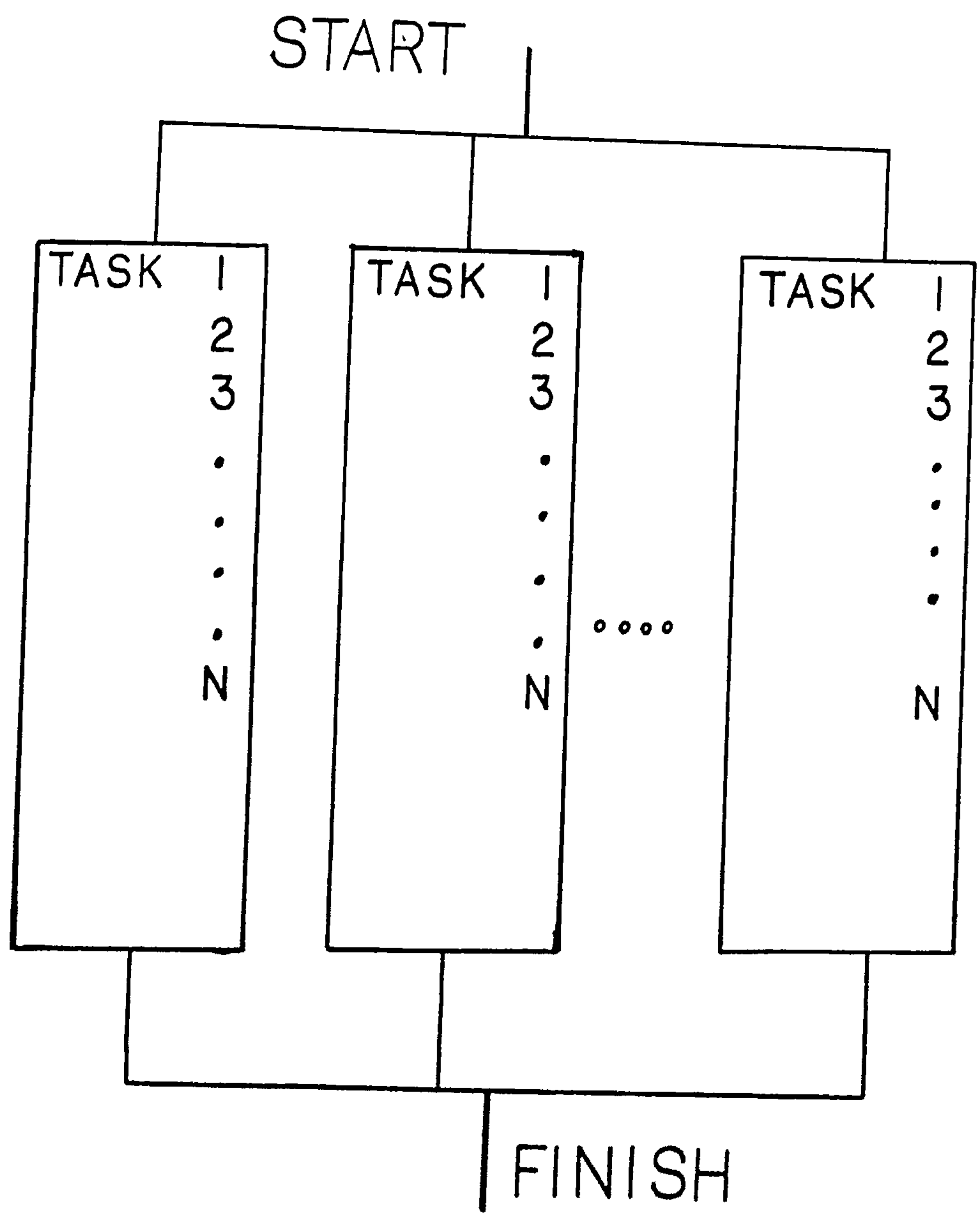


Fig. 5-2: Parallel Configuration

different workstation. The workstations within a parallel configuration use advanced high capability robots to perform multiple tasks at a single workstation.[6] Workstations in a parallel configuration are much more complex than the workstations in a serial configuration because each workstation performs all the tasks assigned to the cell. The basic distinction is that serial lines use many low capability robots to do single tasks while parallel lines use a few high capability robots to do all the tasks. Of course this is the extreme and many times it will be beneficial for each parallel section in a parallel configuration to use more than one robot just as it may be beneficial for workstations in a serial line to be assigned multiple tasks.

Parallel lines produce finished parts at rate equal to the product of the number of parallel workstations and the rate of a single work station. An assembly cell in a parallel configuration can continue to operate as long as there is at least one operable workstation.

There are advantages and disadvantages to both configurations. Part transport requirements are greater with serial configurations and parallel configurations require more expensive robots with advanced capabilities. Soldner and Spitznagel demonstrate that the major advantage of parallel assembly lines is the flexibility to increase and decrease capacity in small increments consistent with requirements.[6] This advantage is realized for products where demand begins low, increases until it remains constant for some time, and then drops off at the end of the product's life cycle, see figure 5-3. In contrast, serial lines do not facilitate small incremental increases and decreases in capacity. When demand is less than the level designed for, serial lines are under utilized. Serial lines tend to be better suited for products that exhibit constant demand throughout the product life cycle.

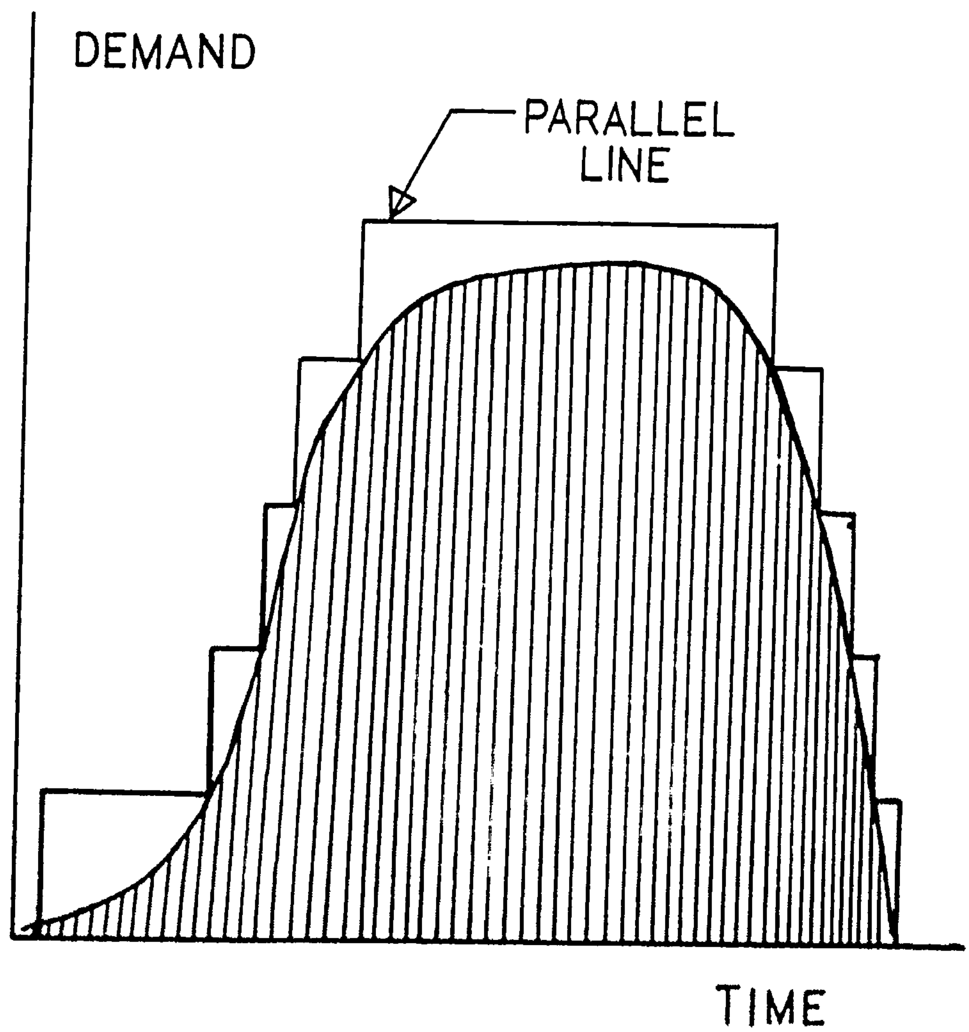


Fig. 5-3: Cell Configuration & Product Life Cycle

6. A ROBOTIC ASSEMBLY CELL

An automated assembly cell was studied as part of this research toward developing a cell level control structure for automated assembly cells. The cell was examined in detail along with the parts targeted for assembly in the cell. The existing manual assembly operations were also studied. This chapter provides a detailed description of the automated assembly cell in an effort to identify the issues important to effective cell control, and at the same time provide an illustration of one type of assembly cell being developed today.

6.1. RESEARCH APPROACH

The study of the assembly cell project began with an attempt to gain a thorough understanding of the product, the existing assembly process, and the proposed assembly cell. A list of questions was prepared whose answers would begin to provide the basic information required to identify and understand the important issues in the design, development, and implementation of the assembly cell. The following are examples of some of the questions:

1. What is the existing assembly process?
2. Is there any tooling used in the assembly process and if there is, what specific function does it provide?
3. What are the quality assurance measures taken by the operators doing the manual assembly?
4. What kinds of problems do the operators experience during assembly?
5. What do the operators do to overcome assembly problems?
6. Are there operator actions which aid in the assembly operation but which would normally not be identified as part of the assembly operation?
7. What are the quality levels of the subassemblies and component parts?
8. What kind of component part inspection is performed and who does it?
9. What subassembly and component part characteristics are inspected/tested?
10. What, if any, changes to the product are planned to facilitate the implementation of automated assembly?
11. Will existing tooling be used in the automated

assembly cell and if so, how will it have to be modified?

Numerous plant visits were made in order to interview the plant personnel who manufactured and assembled the parts. Data was gathered concerning the details of each of the assembly operations as well as the fabrication of certain subassemblies and components which are critical in the final assembly process.

Each engineering function connected with the product was interviewed, this included the plant engineer at the site where the product is being assembled, the manufacturing engineering staff, and the engineer who has responsibility for the product. Quality assurance engineers were also interviewed and the test/inspection processes for the product and its component parts were investigated. Operators were questioned about the assembly, and time was spent observing the operators as they performed the manual assembly operations.

6.2. PART FAMILY

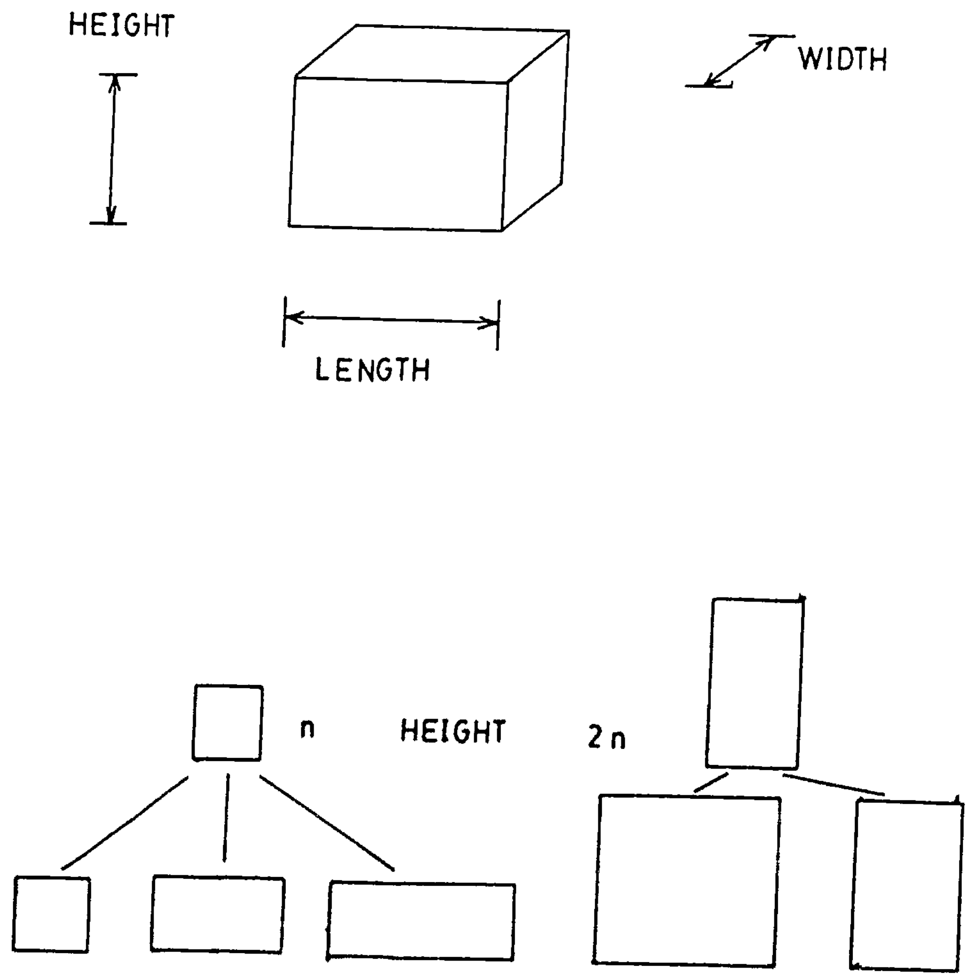
The automatic assembly cell is being developed to do the final assembly for a part family where because of the variations in subassemblies, there are 10 part numbers for each physical configuration. At present final assembly of this part family is manual.

6.3. PART STRUCTURE

Relative to the final assembly, the various parts in this family are differentiated dimensionally. Part height is the first dimension which distinguishes between various part configurations. Part height is some multiple, n, of the fundamental part height and the final assembly operation is different for each configuration. The second dimension which differs among various configurations is part length. For each part height there are m part lengths. The width of the part is common in all part numbers. Figure 6-1 gives a graphical representation of the important dimensional variations within the part family relative to the final assembly.

6.4. PART COMPONENTS

Each part configuration is essentially a stacked assembly. Assembly begins with j subassemblies and k component parts are added along one axis.



PART LENGTH VARIATIONS

Fig. 6-1: Part Family Dimensional Variations

For example, the the final assembly of one n height part consists of:

(A) retainer top
(B) insert, top
(C) n height subassembly
(D) insert, bottom

On some part numbers fasteners are applied to the assembly to enhance its functionality.

Both inserts, (B) and (D), have holes which mate with features of the subassembly (C). The insert (B) is held in place by the retainer (A) which fits over the insert and is joined to the subassembly (C) in a metal forming operation. There are m variations of inserts (D) which are joined to the subassembly. In one part a friction fit is used to join the insert (D) to the subassembly (C). In another part the insert (D) is fixed in place in an additional operation performed after final assembly.

6.5. MANUAL ASSEMBLY

The final assembly operation for the part family is manual. Operators assemble the part by hand. Semiautomatic tooling is used for the metal forming operation required to join a retainer to the subassembly. Another semiautomatic tool is used to apply the fasteners when needed.

Subassemblies (C) come to the final assembly bench in trays. The subassemblies are placed in the tray after the prior operation and are oriented so that all the tops of the subassemblies are facing the same direction. There are no alignment features built into the design of the tray and the subassemblies are free to move. The other component parts, insert (B), retainer (A), and insert (D), come to the assembly bench lose piece in boxes or bags. The assembly steps for the n height part described earlier are:

1. Align the insert (B) with the top of the n height subassembly (C) and insert.

2. Slide the retainer (A) over the insert (B) and onto the subassembly (C).

3. Invert the assembly.

4. Align the insert (D) with the bottom of the n height subassembly (C) and insert.

5. Use the semiautomatic forming tool to join the retainer (A) to the n height subassembly (C). The tool also seats the insert (D) into

position.

6. Use a semiautomatic tool to apply the fasteners to those parts that require them.

6.6. PART MARKING

Part marking will be included as one of the operations in the automatic assembly cell. The amount of information which is printed determines how many sides of the part will be marked. Some parts can accommodate two lines of print on one face but other parts requiring two lines of print must be marked on opposite sides. Some of the smaller parts require marking on four sides but this is only done when the print will not fit on the sides along the length of the part. The automatic assembly cell is being designed with the capability of marking parts along their length. Those parts which require marking on the ends of the part will have this marking done as an additional operation outside the cell.

6.7. OPERATOR INTRINSICS

In all but the most simplistic manual operations there are actions performed by operators which make the work easier, improve productivity, and often enhance product quality, but which were not identified when the operation was defined by the process or methods engineer. While studying the manual assembly operation for this part family it was our objective to identify those kinds of actions in order that their impact on the automatic assembly process might be analyzed. These kinds of operator adjustments will be called "operator intrinsics". The following is a list of operator intrinsics which were observed.

1. When mating an insert with a subassembly the operator often jiggles the insert until the features of the subassembly are aligned with the holes in the insert so that insert can be slid into position.

2. Frequently the operator is unable to align the holes in a insert with the features in a subassembly. When this occurs the operator examines the subassembly and if possible realigns those features which are visibly out of position.

3. During assembly the operator obtains component parts from bags and boxes where they are stored lose piece.

When the operator picks up a component part an automatic visual inspection is done. If there are obvious flaws in the component the component is discarded.

4. At some point in the manufacture of subassemblies some of them are bent or twisted. The operator detects an irregular subassembly when inserting an insert. If the subassembly is bent or twisted the operator rejects the subassembly and places it in a rework or scrap bin. Sometimes the operator may attempt assembly with several inserts before deducing that the problem is an irregular subassembly.

5. During the marking operation, the operator visually inspects each part and as a result, is able to detect and correct problems. For example, if there was not sufficient ink on the print head it is immediately apparent to the operator who can take the necessary corrective action.

6. The fasteners are fed to the applicator from a vibratory bowl through a feed track. The fasteners can jam in the feed track and starve the applicator. The operator detects and corrects feed track jams.

The operator is also aware of parts which do not receive fasteners, even though the applicator cycled, and is able to assure that each assembly has the correct number of fasteners.

6.8. EXISTING / POTENTIAL ASSEMBLY PROBLEMS

A potential problem for automatic assembly is the insert, which must mate with features in the subassembly. The operator's hand-eye coordination and sensory capabilities are hard to duplicate. The ability to jiggle the part in order to achieve alignment is difficult to do with automatic equipment although similar problems have been solved in printed circuit board population [13]. The fact that the subassembly features are sometimes misaligned further complicates this problem.

As a result of the development work on the automated assembly cell, modifications to both parts have been investigated. Changes to the subassembly were not possible, but changes to the inserts have been prototyped. At the time of this writing the results of tests on the modified insert design were not available.

6.9. AUTOMATIC ASSEMBLY

An automatic assembly cell is being developed to do the final assembly and marking of the part family described earlier in this paper. The major components of the cell are five programmable robots, an asynchronous conveyor transfer system, component parts feeders, five special purpose work stations, and a cell control computer and software system.

6.9.1. ROBOTS

Five SCARA (selective compliance assembly robot arm) class robots will be used in the cell. These robots are fully programmable and have their own controller. The controller is a microprocessor running under a single tasking operating system. The robot can communicate with other devices through a number of interfaces; an RS232 port, 24 digital input/output lines, a printer interface, and a cassette interface.

6.9.2. ASYNCHRONOUS CONVEYOR

An asynchronous conveyor transfer system will be used to transport the assembly from station to station within the cell. The conveyor system is configured from six independent straight sections each with its own continuously moving drive belt. Pallets will be used to carry the parts on the conveyor. All turns in the conveyor system are 60 degrees and no input/output control is required for a pallet to make a turn. The pallets are positioned at a work station with a "v" block which locates a pin protruding from the bottom of the pallet. Each pallet is designed to accommodate two assemblies.

6.9.3. SPECIAL PURPOSE WORK STATIONS

Five special purpose work stations will be used in the automatic assembly cell. The first station will apply the primer required for part marking. The second special purpose station will join the retainers to the subassembly in a metal forming operation. The third special purpose station will be a print head used to mark the parts. The fourth station will be a drying tunnel and the fifth special purpose station will apply the fasteners.

6.9.4. COMPONENT PARTS FEEDERS

One feeder bowl design will be used to feed all retainers regardless of length. Three feeder bowl designs will be required to feed all the various inserts. The subassemblies will be palletized in a prior operation and will be fed to the assembly cell with a pallet stacking mechanism. In all of these feeder orienter systems additional sensors will be used to sense part orientations not insured by the feeding system. It was more economical to sense part orientation and correct with a robot than to have the complete orienting capability designed into the feeder bowl systems.

6.10. CELL LAYOUT

The cell is made up of eight stations positioned along a hexagonal conveyor. The conveyor system will be closed and pallets will circulate asynchronously from station to station around the cell (see figure 6-2).

The robots in the assembly cell will grip all components and subassemblies across their width because width is the common dimension among all parts in the part family. The part marking operation is the only time when the part will be gripped across its length.

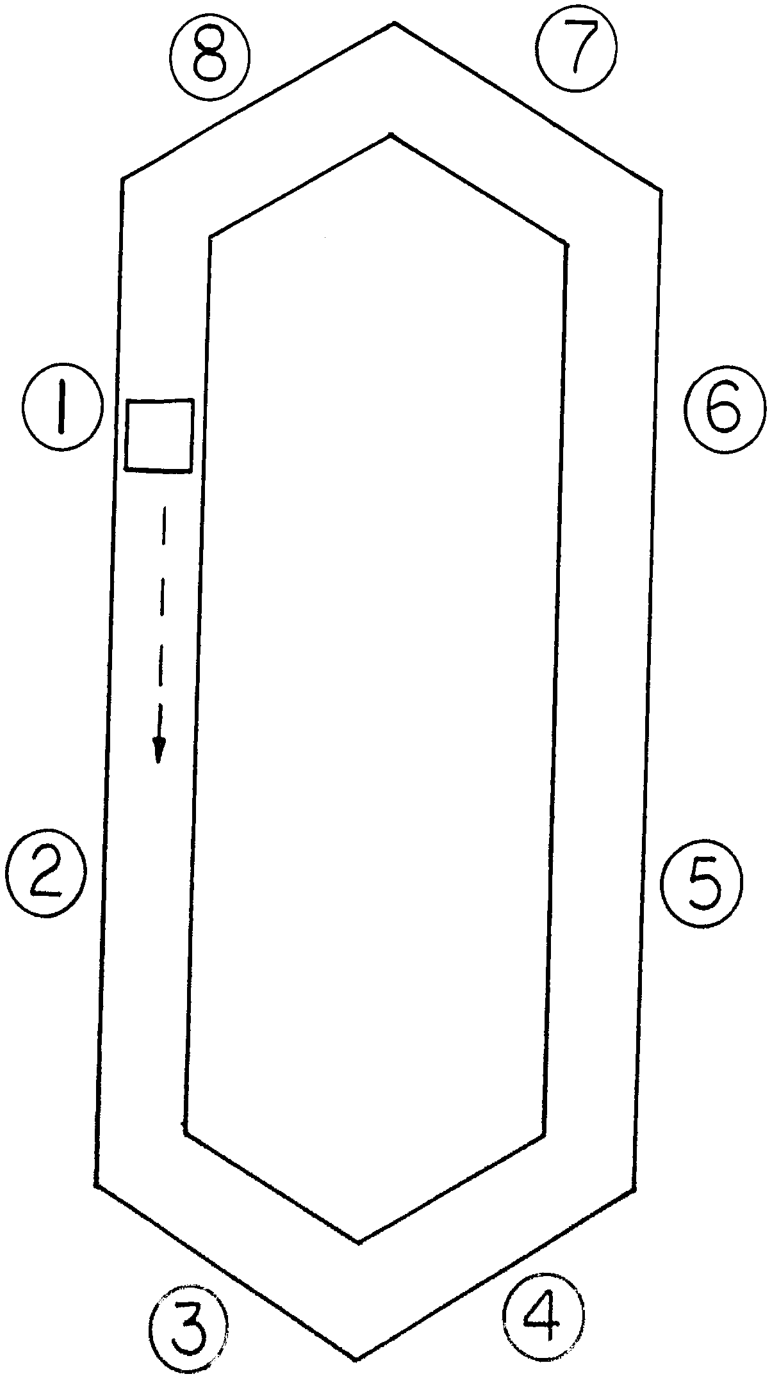


Fig. 6-2: Cell Layout

6.10.1. STATION 1

Station 1 begins assembly by placing subassemblies (C) on the transfer system pallet.

6.10.1.1 STATION 1 COMPONENTS

Station 1 contains the following equipment:

* 1 SCARA class robot
* pallet stacking system
* orientation sensing system

Subassemblies are palletized in the operation prior to final assembly. The pallet stacking system is used to feed the pallets of subassemblies to station 1 during final assembly.

6.10.1.2 STATION 1 TASKS

Step 1. The pallet is stopped and located.

Step 2. The robot moves to the pallet of subassemblies supplied by the pallet stacking system and picks up a subassembly (C).

Step 3. The robot presents the subassembly to the orientation sensor and reads the sensor. If necessary, the robot rotates the subassembly 180 degrees (wrist rotation).

Step 4. The robot moves to the transfer system pallet and puts the n height subassembly into the pallet fixturing.

Step 5. The pallet is released to the next station.

6.10.2. STATION 2

Station 2 is a special purpose station where the priming agent is applied to the part. The part travels on the conveyor with its length parallel to the direction of travel. The priming agent will be applied with a pneumatic applicator and the pallet will not be required to stop. The applicator will brush against the part and dispense the primer as the part moves past the station. The station will be capable of applying primer to both sides of the part at once.

6.10.3. STATION 3

Station 3 is an assembly station. An insert (B) and retainer (A) are added to a subassembly (C) at this station.

6.10.3.1 STATION 3 COMPONENTS

Station 3 contains the following equipment:

* 1 SCARA class robot
* 2 insert feeder/orienter systems
* 1 retainer feeder/orienter system

6.10.3.2 STATION 3 TASKS

Step 1. The pallet is stopped and located.

Step 2. a) The robot reads the orientation sensor to determine the presence and orientation of the insert (B) in the feed track.

b) The robot picks up the insert (B) and moves to the pallet. During the move the insert is rotated 90 degrees to its proper orientation.

c) The robot fits the insert (B) onto the subassembly (C), already in place on the pallet.

Step 3. a) The robot reads the orientation sensor to determine the presence and orientation of the retainer (A) in the feed track.

b) The robot picks up the retainer (A) and moves to the pallet. During the move the retainer is rotated 90 degrees to its proper orientation.

c) The robot places the retainer over the insert (B) and onto the subassembly (C), both of which are already in place on the pallet.

Step 4. The pallet is released to the next station.

6.10.4. STATION 4

A special purpose metal forming machine at station 4 joins a retainer (A) to a subassembly (C). The assembly remains in the transfer pallet fixturing during the metal forming operation.

The task sequence at station 4 is:

Step 1. The pallet is stopped and located.

Step 2. The forming machine cycles.

Step 3. The pallet is released to the next station.

6.10.5. STATION 5

Station 5 performs the marking operation. The station is being designed to print up to two lines. The two lines may be printed on one side of a part, or one line on each side of a part.

6.10.5.1 STATION 5 COMPONENTS

Station 5 contains the following components:

* 1 SCARA class robot
* 1 automatic print head

In addition to the above equipment special purpose tooling is being designed to support the robot arm while the part is being printed. Because the robot must grip the part across its length and because the part lengths vary significantly the station is being designed so that the robot will be capable of changing its own gripper.

6.10.5.2 STATION 5 TASKS

Step 1. The pallet is stopped and located.

Step 2. The robot moves to the pallet and picks up an assembly.

Step 3. a) The robot moves to the marking machine and presents the area to be printed. Tooling moves into place to support the robot arm and the print head cycles.

b) If a second line of print is required the robot rotates the part 180 degrees (wrist rotation) and presents the opposite side of the part to the marking machine at a new location. Then the support tooling moves into place and the print head cycles again.

Step 4. The robot returns the part to the pallet.

Step 5. The pallet is released to the next station

6.10.6. STATION 6

Station 6 is designed to cure the ink used to mark the part. This pallet will pass through the tunnel and will not be required to stop. The station design was not finalized at the time of this writing and conr!deration has also been given to stopping the pallet within the tunnel. Nevertheless neither method appears to present any problem.

6.10.7. STATION 7

Station 7 applies the fasteners to those parts requiring them. When the cell is assembling those parts that do not require the fasteners the station will be inactive.

6.10.7.1 STATION 7 COMPONENTS

Station 7 contains the following equipment:

* 1 SCARA class robot
* 1 automatic fastener applicator
* 1 fastener feeder/orienter system
* 1 part inverter

Some parts must be inverted in order to apply all the required fasteners to the part. Due to the way the part is gripped and the fact that SCARA class robots are being used it was necessary to design a simple part inverter.

6.10.7.2 STATION 7 TASKS

Step 1. The pallet is stopped and located.

Step 2. The robot moves to the pallet and picks up an assembly.

Step 3. The robot moves to the fastener applicator and and presents the assembly.

Step 4. The fastener applicator cycles and applies the fasteners.

Step 5. The robot removes the assembly from the fastener applicator.

a) if the part must be inverted for the application of additional fasteners, the robot moves to the inverter, and presents the assembly to the inverter.

b) The inverter inverts the assembly.

c) The robot removes the assembly from the inverter, moves to the applicator, and presents the assembly to the applicator.

d) The fastener applicator cycles and applies the fasteners.

e) The robot removes the assembly from the fastener applicator.

Step 6. The robot moves to the pallet and places the assembly back on the pallet.

Step 7. The pallet is released to the next station.

6.10.8. STATION 8

Station 8 is the final assembly station. Station 8 adds the insert (D) to the assembly.

6.10.8.1 STATION 8 COMPONENTS

Station 8 contains the following equipment:

* 1 SCARA class robot
* 1 retainer feeder/orienter system
* 2 insert feeder/orienter systems

6.10.8.2 STATION 8 TASKS

Step 1. The pallet is stopped and located.

Step 2. The robot picks up the partially completed assembly (A,B,C) which is on the pallet.

Step 3. a) The robot reads the orientation sensor to determine the presence and orientation of the insert (D) in the feed track.

b) During the move to the insert (D) feeder track, the partially completed assembly is rotated 90 degrees to its proper orientation.

Step 4. The robot mates the partially completed assembly with the insert (D).

Step 5. The completed assembly is moved to the finished part tote and placed in the tote.

Note: For parts where the insert (D) is not held in position with an interference fit, the robot must regrip the completed assembly on the insert (D) before moving to the finished part tote.

Step 6. The pallet is released to the next station.

7. CELL CONTROL

As described earlier, typical assembly cells are characterized by the use of intelligent programmable devices, the use of computer control at the station/device level, the use of sensors, the interconnection of devices for communication, and by a cell level interface to human operators and other factory systems. A typical assembly system is comprised of parts feeders, orienters, robots, material handling systems, and special purpose work stations. Such assembly cells require a control system which provides the level of integration necessary to achieve a synergistic effect resulting in significant real productivity increases.

In this chapter a cell control architecture is developed for automated assembly cells. This architecture is general enough so that its basic structure can be applied across a larger class of manufacturing cells. The architecture will result in systems which are easily modified and enhanced, and the architecture enables the development of control systems in which the software / hardware boundary is transparent.

7.1. SCOPE

Before a control system architecture can be developed its objectives must be defined relative to its environment. Various approaches to specific applications have been documented.[21,22,23,24,25,26] The majority of these control approaches are system specific and fail to define a general control architecture capable of being applied to a whole class of systems.

This chapter is concerned with defining a general control system architecture for an automatic assembly cell. This architecture is being defined for cell level control. In the context of cell level control, such a system provides the cell level interface to other factory systems and human operators. The control system also serves as the integrator of the individual devices which make up the cell. The cell level control system does not directly control intelligent programmable devices such as robots or NC machines. These devices have their own machine controllers. The cell control system interfaces with the device controller and provides the next higher level of systems control.

7.2. ASSEMBLY CELLS & MACHINING CELLS

Automated assembly cells differ from automated machining cells and this difference is reflected in their control systems. Automated machining cells typically consist of several NC machine tools, one or more robots, and possibly a material handling system. If properly designed it is possible to have different NC machines tools machining different parts within a part family at the same time. Given this capability one of the most important jobs of the control system is to effectively schedule and load the machines in the cell so that machine utilization is maximized while achieving the production schedule.

The cell components in an automated assembly cell are much more closely coupled than the machine tools in a machining cell. In assembly cells the assembly process is usually distributed among two or more devices that perform assembly operations on a part as it moves in a fixed sequence from station to station. Within an assembly cell the assembly process, once begun, is usually completed without interruption. Whereas in machining cells, it is often possible to interrupt the sequence of operations on one part, perform some number of operations on another part, and then complete the remaining operations on the first part. This is accomplished through the use of cell scheduling and loading systems that attempt to achieve high equipment utilization while meeting due dates. As a result there is more emphasis on device (machine tool) loading in machining cells than there is in assembly cells. Assembly cell equipment utilization is impacted more by line balancing.[14] Cycle times are also different. The cycle times at each station in an assembly cell are generally on the order of seconds as compared to much longer cycle times found in machining cells.

7.3. NEED FOR CELL CONTROL

Programmable automation (flexible automation) has been developed because the high cost of special-purpose automation equipment is not justified for batch manufacturing. Automated assembly cells arise out of the need to improve the productivity of batch assembly. Typical automated assembly cells bring together programmable robots, material handling devices, and other special purpose equipment to form an assembly system for a part family or group of similar parts.

The interactions that result from bringing together multiple devices and systems to form an assembly cell, require coordination and control. With integration comes interdependency and with interdependency comes the need for control. Once a robot or other device becomes part of an assembly system its actions begin to impact other members of the same system. The need for external control arises out of each stations lack of data. It is unwise, and would be a waste of resources, to design a system so that each system component is aware of the system's status at any point in time and is capable of predicting the impact of its actions on the system. Therefore, there is a need for cell level control.

A cell level control system combines the flexibility of each of the cell components to form an automated assembly cell which is flexible and adaptable. The cell level control system is the "integrating agent" which transforms a group of interrelated devices into a flexible assembly cell.

Integration, although a vitally important ingredient, is not the only consideration in the development of a control system for an assembly cell One of the far-reaching objectives in the development of automated assembly cells is autonomy. Many of the economic decisions which surround the development of automated cells are based to some degree, on autonomous cell operation. Reductions in manual assembly and the resulting labor savings are often used to justify the expenditures for cell equipment. "Autonomous" implies more than automatic.[15]

For a cell to operate independently it must be tolerant of errors. This means the cell must be able to detect errors and take the appropriate action to recover. If the cell is intended to operate unattended for significant periods of time it must be capable of handling complex error states. In the past, cell control was thought of as the sequence control which provided the "lock step" device coordination common in many early cells, but the scope of cell control must be expanded. In this context, a cell control system not only controls the cell but manages it. Incorporated in a control system should be functions such as; operations control, error detection and recovery, reporting, device programming, and safety control. The functional elements of a control system will be discussed more fully later in this chapter.

7.4. THE NBS HIERARCHICAL CONTROL STRUCTURE

The National Bureau of Standards, Industrial Systems Division has developed a theoretical hierarchical control structure.[16] This control rationale is not limited to a cell but is intended to be applied through several levels of manufacturing systems. Several important concepts which underlie the design of this hierarchical control system will be examined in this section.

NBS has also used the concepts of hierarchical control to design and implement a real-time control system for a robot.[17,18] These efforts have become the foundation of the control systems being developed as part of the Automated Manufacturing Research Facility project at NBS.[19,20]

The NBS hierarchical control system is a control rationale based on task decomposition, modularization of control, and standard interfaces. To this point the majority of the work has been directed toward the development and implementation of a hierarchical control system for robot control.

7.4.1. A MULTI-LEVEL STRUCTURE

The hierarchical control system is based upon a multi-level structure. At each level in this multi-level control structure, tasks are either processed or decomposed into sets of subtasks which are supplied to the next level of the hierarchy. This continues until the bottom level of the control structure is reached and primitive commands are output to the device or system being controlled.

Each level of the NBS hierarchical structure is actually partitioned into three separate segments, as shown in figure 7-1; (H) a goal or task decomposition module, (G) a feedback processing module, and (M) a world model module. At all levels in the hierarchy H, G, and M modules are concurrent processes produced by the simultaneous execution of real-time programs. The H module at each level of the hierarchy performs the task decomposition function. The H module receives goal oriented commands or tasks from the next higher level of the hierarchy and either initiates some process at the level or decomposes the task into subtasks which are provided as input to the next lower level in the hierarchy.

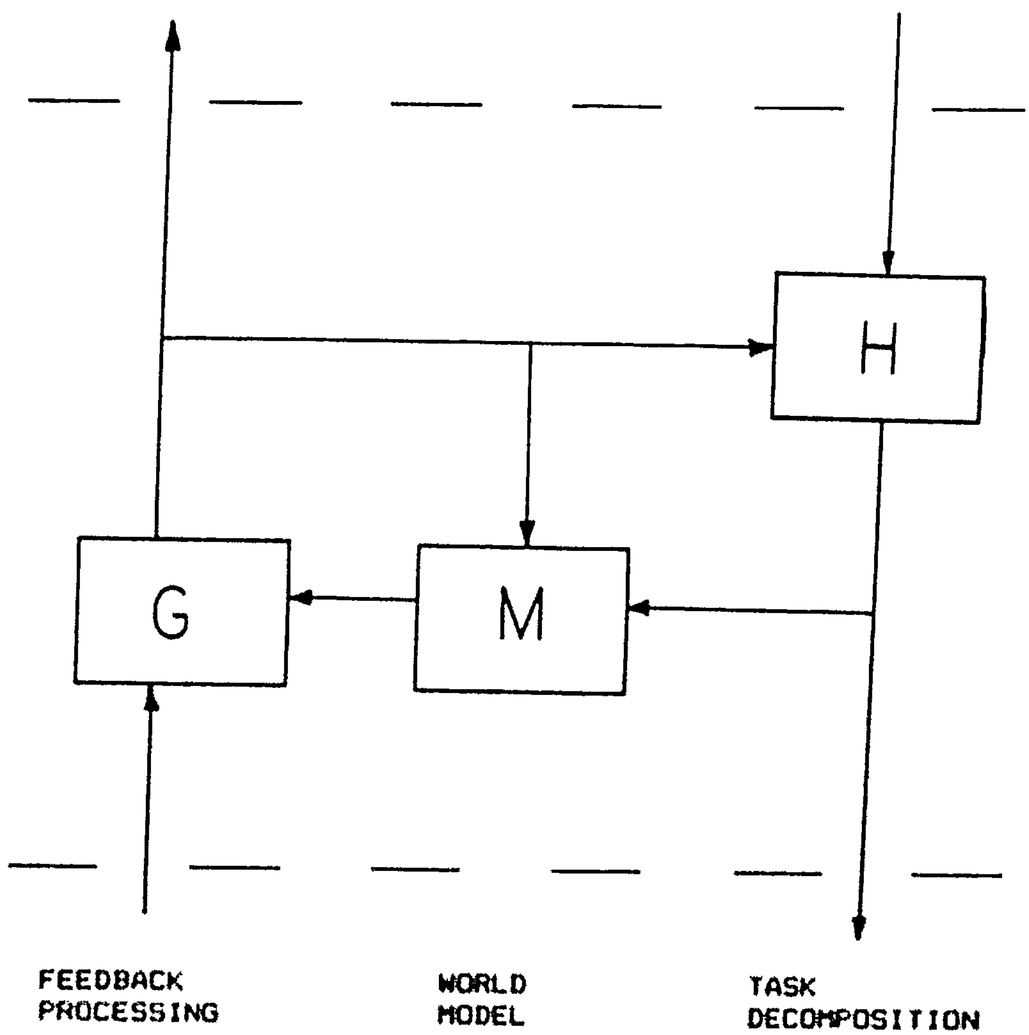


Fig. 7-1: Partitioning At A Control Level

Each level of the hierarchy is serviced by a feedback processing module G. The feedback processing module processes the sensory data stream and the feedback information from lower level control modules in order to provide the information necessary for control decisions.

The world model module at each level in the hierarchy is used to supplement the sensory and feedback data. A world model module consists of a knowledge base containing all the information currently known about the world which is relevant to the particular level of the control hierarchy. The knowledge base may contain information about the task, the parts being manipulated, and the workplace. This information is used to generate expectations or predict the sensory data which is being provided by the feedback module. If the predictions compare favorably with the actual sensory data then the auxiliary information contained in the world model, but not available from the feedback system, can be used in making control decisions.

7.4.2. FUNCTIONALLY BOUNDED MODULES AND STATE TABLES

A functionally bounded module is described as being the atomic unit within a control level. A functionally bounded module is defined as a set of inputs, a process, and a set of outputs. Functionally bounded modules can be grouped together under a single unit called an "executing owner". Both functionally bounded modules and executing owners are given names which are used to invoke them within a control level.

The structure of a control level is generic and standard for each level in the hierarchy, see figure 7-2. A control level consists of three parts; a preprocess function, a state table function, and a post process function. The preprocess function transforms input data into a form suitable for the control level. The state table performs the decision processing for the control level. The postprocessing function transforms outputs into forms suitable for other systems.

Task decomposition at each control level is accomplished through the use of a state tables and the procedures (implemented as functionally bounded modules) called by the state tables. A state table is based on a set of conditions. These conditions use the input variables and the internal state of the control system to test the relevant state of the world. A state is defined as a

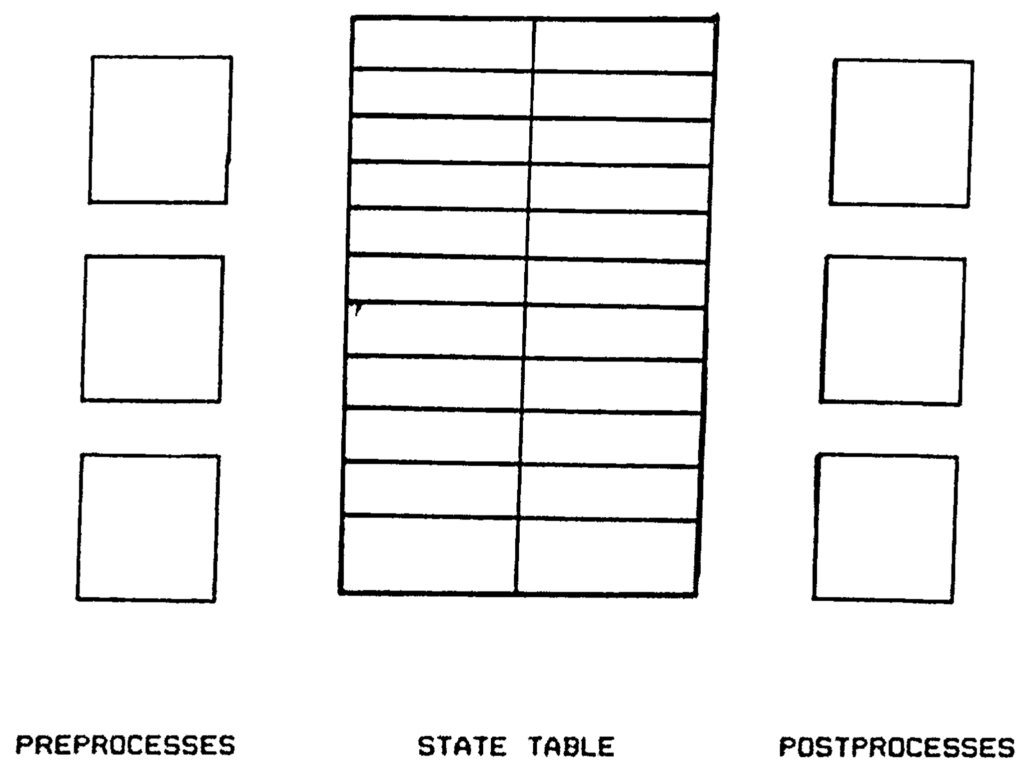


Fig. 7-2: Control Level Structure

unique set of values for the conditions. There is one line for each state in the state table, see figure 7-3. Each line corresponds to a relevant state of the world. A line contains the values for the conditions defining the state and the list of output procedures which are executed. Each input command at a control level has a state table which is executed whenever that command is received from the level above. In one implementation of this structure at NBS, twenty commands and their corresponding state tables were implemented at a control level.

NBS elected to implement the decision making at each control level in state tables. They saw several characteristics of state tables as advantages over procedural software programs. Because the state table inputs are examined during every cycle, the NBS system avoids the use of interrupts. When a particular condition requires immediate attention it can be indicated in the state table and attention can be initiated within one cycle time. State tables provide a convenient mechanism with which to handle new conditions. As new conditions or states are identified the state table is expanded by simply adding a new line, and the state table processing

STATE TABLE

STATE #	CONDITION 1	CONDITION 2	CONDITION 3	CONDITION 4	ACTION
1	TRUE	FALSE	0 < X < 50	TRUE	CLOSE VALVE
2	TRUE	TRUE	51 < X < 100	FALSE	OPEN VALVE
3	FALSE	TRUE	0 < X < 50	TRUE	HEATER ON
4	FALSE	TRUE	51 < X < 100	FALSE	HEATER OFF

Fig. 7-3: Example Of A State Table

system does not need to be changed. In this way the system developers need only be concerned with what action must be taken on certain conditions, and need not be concerned with when and where these conditions might occur within a procedural structure.

7.4.3. COMMUNICATION

The concept of a "common memory" is used to implement communication. Modules write to "mail boxes" without regard to intermodule protocols. A standard communication process is used to hide the details of a transfer from the application state tables and procedures. All that is necessary is the symbolic name of the destination and the communication process handles the rest. What has been implemented is a standard communications handling system at the application software level which hides the details of physical addresses and protocol.

7.4.4. ASSESSMENT

Several important concepts underlie the design of this hierarchical control system. Modularization is emphasized in the development of each level of the hierarchy. Each level is independent and levels communicate via standard interfaces. This simplifies design and makes it possible for modifications to one level of the hierarchy without requiring corresponding changes in other levels. The notion of task decomposition is generally accepted as a useful tool in the simplification of complex systems and serves as the basis of the NBS hierarchical approach.

The use of standard interfaces reduces the intermodule coupling between control levels in the hierarchy and supports modularization. This approach permits the implementation of change within a module without generating any unwanted side effects in other modules. The communication handling system which also makes use of a standard interface, reduces complexity by hiding the details of communication, such as protocols and physical addresses, from the application level. These approaches are generally accepted and widely endorsed in systems design.

An integrated sensory processing and abstraction subsystem at each control level provides the necessary transformation of raw data into useful feedback while shielding the control function at each level from the details of sensor control and sensory data integrity. As autonomy becomes one of the major design objectives in automated cells, the cell control systems will be forced to process, organize, and abstract an increasing variety of sensor data from a variety of sources. A hierarchical approach to the manipulation of sensor and feedback data is a logical extension to a hierarchical control methodology.

7.5. A CELL LEVEL CONTROL STRUCTURE

Cell level control can be viewed as consisting of four layers; (1) the interface layer, (2) the cell management layer, (3) the execution layer and (4) the device dependent layer (see figure 7-4). The interface layer controls all interaction between the cell and the human operators as well as any interaction with other factory systems. The cell management layer manages the cell's resources in order to carry out the high level tasks generated at the interface layer. The execution layer provides the coordinated run-time control of the cell's components. Finally, the device dependent layer contains the device drivers necessary to implement the logical device commands, which are generated at the execution layer and which direct cell operation.

I N T E R F A C E   L A Y E R

C E L L   M A N A G E M E N T   L A Y E R

E X E C U T I O N   L A Y E R

D E V I C E   D E P E N D E N T   L A Y E R

Fig. 7-4: Multi-Layer Control Structure

Task decomposition and hierarchical control are important concepts embodied in this control structure. A layered approach which couples task decomposition and modularization is used to develop a control system structure which results in control systems which are easily maintained and modified. The use of task decomposition and basic hierarchical control is similar in many respects to the hierarchical control strategy being developed at NBS. The concept of standard interfaces is also used. Transparency in the form of "black boxes", where the inputs, process, and outputs are well defined, but where implementation is hidden, is also an important system design concept.

One of the ways that this control structure differs from the NBS approach is that functional modules can defined within hierarchical layers. Functional modules are used to reduce complexity. A single functional module processes a closely related subset of the layers tasks. These tasks are said to be "functionally related". Functional modules can exist at any layer and are unique to a layer. In other words the functional modules within different layers are different. The use of functional modules increases the flexibility and adaptability of a control system. As new and improved techniques are developed, obsolete functional modules can be replaced without perturbing the system as long as standard interfaces are maintained.

7.5.1. INTERFACE LAYER

Each layer has been decomposed into functional modules (see figure 7-5). The interface layer provides an interface between the cell and the human operator, and it also provides an interface between the cell and other factory systems. The operator interface is a significant factor in determining the size of the productivity increases realized through the implementation of automated assembly cells. As a result, the operator interface should be carefully designed for ease of use so that the full potential of the cell is achieved. The interface should be designed around accepted human factors principles so as to efficiently utilize the operator's skills without creating an excessive physical or cognitive burden.

Lyman and Madni have developed a framework for examining the roles of human operators in robotic systems.[27] Their analysis is also useful in examining the operator's

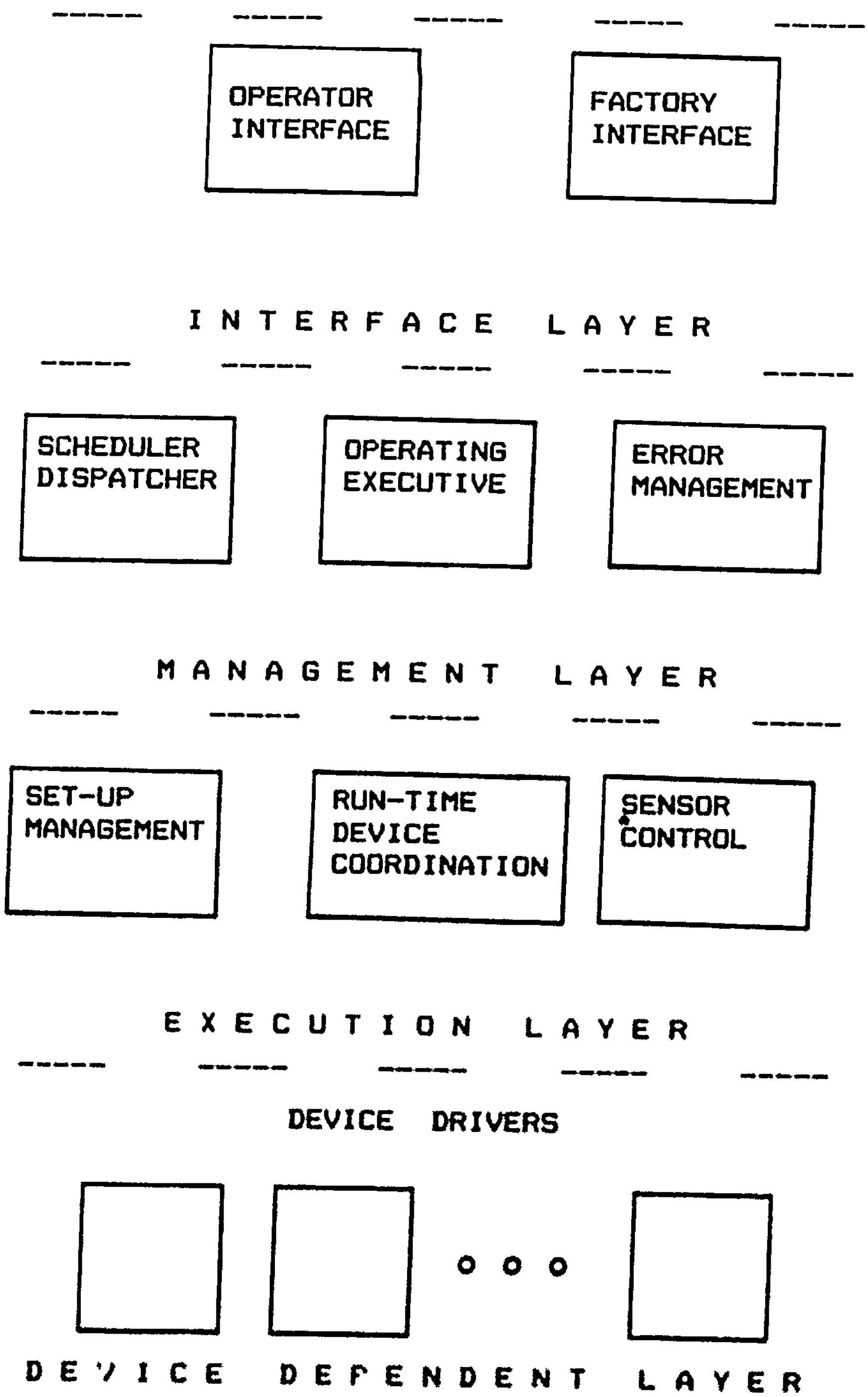


Fig. 7-5: Layered Control Structure - Functional Modules

role in automated assembly cells and other similar systems. They state that operator roles fall under three categories; monitor, manager, and maintainer. Within each of these role categories Lyman and Madni have identified a set of functions common to the particular role, see figure 7-6.

According to Lyman and Madni the monitor reviews the status of the system and intervenes when the task is not progressing as planned. The maintainer functions as a serviceman, repairman, and trouble shooter. The maintainer performs preventative maintenance, identifies problems, diagnoses them, and performs any necessary corrective maintenance.

While in the manager's role, the operator sets goals, supervises the system, organizes and interprets data, controls system operation based on high level goals, and handles contingencies and malfunctions. Planning and goal setting occur at the highest level of the control hierarchy and are the most abstract tasks the operator is faced with. These tasks test both the breadth and depth of the operator's knowledge and experience.

Lyman & Madni's Roles of the Human Operator in Robotics

```
                  ¦---- Watchman
                  ¦
Monitor ------¦
                  ¦
                  ¦---- Task Progress Monitor


                  ¦---- Overall Goal Setter/Planner
                  ¦
                  ¦---- System Supervisor
                  ¦
Manager ------¦---- Data Organizer
                  ¦
                  ¦---- Controller
                  ¦
                  ¦---- Contingency/Malfunction Handler


                  ¦---- Serviceman
                  ¦
Maintainer --¦---- Repairman
                  ¦
                  ¦---- Troubleshooter
```

Fig. 7-6: Operator Roles

The system supervisor sub-function charges the operator with setting schedules and making task assignments. The system supervisor can be forced to revise goals, narrow task assignments, and abort missions, as he monitors the system's operation.

As a data organizer the operator gathers, abstracts, and interprets the data provided by the system. System design has a significant impact on the ease with which the data organizer function is performed. The organization of data should be highly automated so that the operator is only burdened with the high level data interpretation required to support decision making.

Layman and Madni describe the controller sub-function as a command generator. The operator issues a command to the system, evaluates the response, and issues the next command. The contingency handler has two key duties; to anticipate abnormal situations and to respond to system malfunctions. In this role the operator forms contingency plans based on his understanding of the potential system problems.

Several human factors problems have been identified with respect to the operators of automatic systems which include robots:

* employing effective learning approaches to increase operator expertise
* identifying appropriate skill requirements and task burdens
* assessing human performance capabilities against task requirements
* assigning tasks to the operator on the basis of suitable criteria
* assuring health and safety
* assessing the capability of both the trained and untrained operator

Layman and Madni go on to point out, that perhaps the most difficult human factors problem is dividing responsibilities between the operator and, in their case, the robot. They suggest that an appropriate approach to this problem is to break down the task into its functional components, and making tradeoffs where possible. Their list of functional components, which is also applicable to more complex systems, includes:

* sensory modes and levels of stimulation
* data storage

* knowledge encoding and transfer
* programming capability
* operations in failure mode
* adaptability to multiple operation
* span of control by one operator
* endurance limits

Layman and Madni also add that ideally, task allocation should be situation adaptive. The tasks allocated between the system and the operator should shift according to changing demands.

The monitor-manager-maintainer framework provides a useful perspective from which to consider the important issues in the design of an operator interface. The importance of a well designed operator interface to the success of an automated assembly cell can not be over emphasized.

The purpose of the interface layer is to facilitate the necessary interaction between the cell and its human operators, and the interaction between the cell and other factory systems. The monitor-manager-maintainer framework offered some insight into the kind of cell/operator interaction required to manage an automated assembly cell. In a sense the interface layer "connects" the operator to the elements of the cell control structure which provide the capabilities for cell management. The important point is that the nature of this "connection" is such that the operator need not be concerned with the details of the implementation of the cell management functions. The operator is given the capability to issue high level directives or commands without being concerned with how they will be carried out. The parallel to this notion is a management information system (MIS) that gives a manager access to a huge wealth of company data without requiring him to know or understand the complex database structures used to store the data. The important associations are logical. The same is true, although on a smaller scale, for an automated assembly cell interface.

The interface layer of the cell control system provides the operator with a set of capabilities required to manage an automated assembly cell. Cell configuration (sequential or parallel), the complexity of the assembly task, the number of cell components, and the type of transfer system, along with other factors, will determine the relative complexity of the interface layer and the types of capabilities which are required for the operator to effectively manage the cell. These capabilities may include:

* cell power-up and power-down
* cell initialization
* cell configuration
* establish operating parameters
* modify operating parameters
* cell scheduling
* task prioritization
* task initiation
* task suspension and resumption
* task termination
* management of part programs and part program libraries for programmable devices
* management of the tooling and fixtures under cell control
* management of materials under cell control
* remote device control
* reporting of cell status, including:
  - the status of each cell component
  - the names of the control programs executing on the programmable devices
  - the number and type of failures that

occurred in the last time period

- the number of assemblies initiated in the last time period
- the number of assemblies completed in the last time period
- the number of assemblies that were rejected for scrap or rework in the last time period
- the number of each type of component part rejected in the last time period

The interface layer is important to the cell control structure in that it provides a mechanism whereby human operators and other factory systems can communicate with the cell through a single standard interface, regardless of the task. The human operator or other factory system need only be concerned with one standard interface and thereby avoids the complexity involved in interacting with the various subsystems of the cell. The interface layer interprets a task command or request, interacts with the appropriate management layer modules, and provides feedback in such a way that the lower levels of the control structure are transparent. The idea of operator roles, human factors concepts, and standard interfaces are key to the development of an effective control mechanism which supports the management of an automated assembly cell.

7.5.2. MANAGEMENT LAYER

The management layer of the cell control structure consists of modules which manage various cell subsystems and resources in order to carry out the tasks generated by the interface layer. The management layer consists of three functional modules; scheduler/dispatcher, cell operating executive, and error management, see figure 7-7. These modules manage the cell based on a set of high level goals and objectives.

When the management layer receives a command to produce a part, or batch of parts, the scheduler/dispatcher prepares the system. The scheduler/dispatcher interacts with a number of auxiliary modules to verify that the system is prepared to assemble the given part number. Before a part can be assembled there are a number of preproduction requirements that must be met. The scheduler/dispatcher interacts with the program management module to insure that the part programs, associated with the part number, exist for each of the programmable devices required to

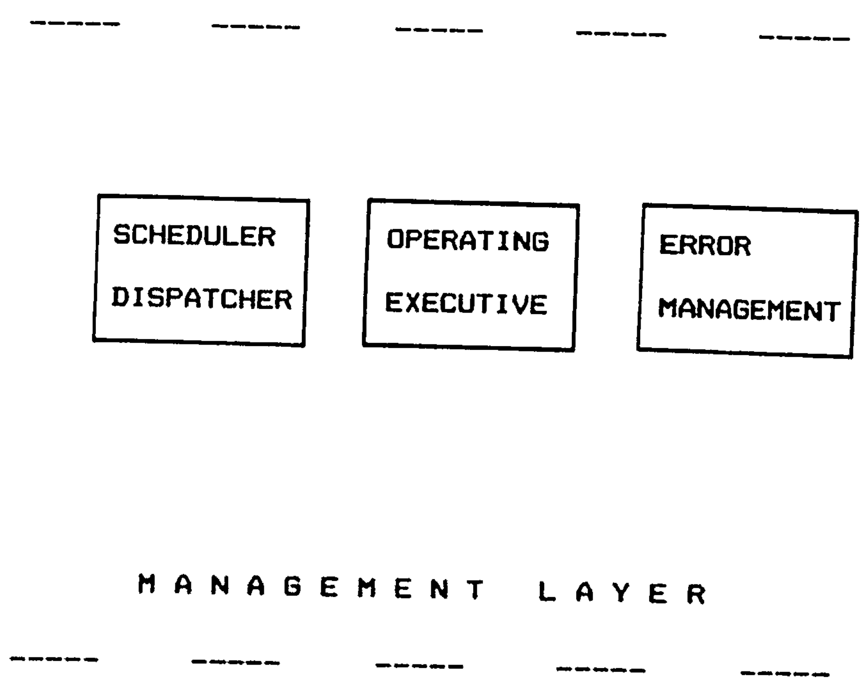


Fig. 7-7: Layered Control Structure - Management Layer

assemble the part. The scheduler/dispatcher must also verify with the materials management module that the required component parts are available. The tooling management module is queried to verify that the tooling and fixtures used in the assembly of the part are also available. Once the preproduction requirements are met the status of the order is updated.

When a preproduction requirement for a particular order is unsatisfied, the order's status is updated and the operator, or other controlling factory system, is notified through the cell interface layer. Generally a request for service is initiated as a result of this situation. Once the service request is satisfied, the scheduler is free to begin the preproduction requirements verification process again.

The scheduler/dispatcher releases orders to the operating executive based upon a set of operating parameters. These operating parameters are designed to minimize set-up times and set-up costs, maximize machine utilization, and meet production due dates. The scheduler/dispatcher uses the operating parameters in combination with the cells current status to release orders to the operating executive. The scheduler/dispatcher may also be overridden by commands from the interface layer.

The operating executive manages the cell's operation by decomposing tasks received from the interface layer. These tasks are processed by the operating executive and subtasks are generated. These subtasks are provided as input to the execution layer. Through task commands to the execution layer the operating executive manages activities such as, cell set-up, normal operation, and maintenance. At the management layer the operating executive works closely with the scheduler/dispatcher and error management modules to manage cell activities. The operating executive is the high level center of cell control.

The error management module detects, diagnoses, and recovers from errors. This module is primarily concerned with the physical errors and faults within the assembly cell and is is not concerned with hardware faults at the device level. Other fault tolerance techniques, such as redundancy, are used to overcome hardware faults at the device level. As a result the error management module is concerned with the physical errors associated with the elements that make up the cell (robots, parts feeders and orienters, transfer mechanisms, and special purpose work stations). These errors include:

* jammed feeders
* empty feeders
* defective components
* parts which are improperly oriented
* joining, mating, and insertion failures
* part slippage in robot grippers
* misaligned fixtures
* damaged or worn tooling

The Robotics Research Group at the University College of Wales in the U.K. is doing research in error recovery in robotic applications. They have taken an approach to error recovery which can be extended, in part, to automated assembly cells.[28] They point out that many errors and faults can be anticipated by systems designers and that contingency features can be engineered into the cell. Nevertheless, given human limitations, there is always the potential for errors which have not been anticipated. But if the system and its operators are to be protected from injury all errors and faults must be detected. Hence the need for robust error and fault management.

Error detection requires detecting devices. It is generally accepted that a range of multi-modal sensors are necessary to cover various contact and non-contact events.[28] In the layered control structure proposed here, sensor information collected from a whole host of sensor systems is necessary for error management. However, the error management module is only concerned with the interpretation and use of sensor data and is not involved in the direct control of sensor systems, this is accomplished in the execution layer.

A sensory driven foward recovery strategy which uses a knowledge base to guide recovery decisions has been proposed by Lee, Hardy, and Barnes.[28] In this approach sensors are used to monitor operation, and normal operating ranges are established for sensory data. Errors are detected when sensory data falls outside of the normal operating ranges. The second stage is error diagnosis. Sensory data is correlated with data from the knowledge base to determine the nature and cause of an error. Detailed analysis of system tasks is used to build fault trees in the knowledge base. These fault trees are used together with the system status information and the sensory data to determine potential error causes. The third stage is error recovery. In this stage error recovery actions and strategies are selected and formulated based upon error diagnosis.

Error recovery systems which select recovery actions from an existing set of predefined methods, although significant, do not go far enough. By their very nature such systems are limited to the set of predefined recovery methods. Remembering that one of the objectives of automated assembly cells is autonomy, it becomes evident that error recovery systems will be forced to synthesize recovery methods dynamically. But before generative recovery systems can be realized, significant amounts of data and knowledge about the cell, its components, the tasks, the parts, and the types of errors must be captured and organized. Using this data becomes a problem very much the same as those problems that are being investigated in the fields of expert systems and artificial intelligence.

Error management will include three basic components; error detection, error diagnosis, and error recovery. How extensive the error management module is, will be determined by the nature of the cell and the complexity of the assembly tasks. Basic error management systems will be primarily concerned with error detection and diagnosis, and will depend on the operator to assist or direct error recovery. As more research is done in this area, sophisticated error recovery capabilities will be developed which will move automated cells closer to autonomous operation.

7.5.3. THE EXECUTION LAYER

The execution layer provides the integration and coordinated device control for the cell. The execution layer consists of three modules; set-up control, run-time device coordination, and sensor control.

Although this layer provides multiple device coordination and integration, the control is achieved in a device independent manner. The execution layer, like the interface and management layers, is device independent. In other words device specific control is not embedded in the execution layer. Device coordination and integration at the execution layer is accomplished by employing the concepts of device independence and logical devices.

The concepts of device independence and logical devices are not new. These concepts have been used successfully in the area of computer graphics.[29,30] One of the problems in the field of computer graphics is software/hardware compatibility. Typical computer graphics application software is developed for a specific type of display device (raster displays, vector displays, pen plotters) manufactured by a specific vendor. A problem arises when one wishes to use computer graphics software with devices for which it was not written. Most often the graphics languages used to drive different types of devices manufactured by different vendors are not compatible. Similarly a robotic program to load a pallet, written in VAL for a Puma robot, can not be used with another vendor's robot which uses a different robotics language. In computer graphics, standardized graphics packages were developed to solve this problem. These graphics packages facilitated the development of graphics systems which could be used on any number of different display devices.

The major concepts which enabled the development of such a standard, were task decomposition and device independence.

The basic approach was to define a set of standard device independent tasks used in computer graphics. This permitted the separation of the device independent task and the device dependent implementation. What resulted were multi-layered device independent graphics packages with standard interfaces between the device independent layer and the device drivers, see figure 7-8. A set of low level commands are provided as input to the device drivers and the device drivers implement the commands in the particular device specific control language used with their device.

A similar approach to cell control is being proposed here. The execution layer integrates and coordinates the cell's devices and work stations through a set of "logical devices" and "execution level primitive tasks". For instance, a "robot" might be one type of logical device. In a particular cell the physical realizations of a logical robot might be a Cincinnati Milacron T3, an IBM 7535, or both. An execution level primitive task for an assembly cell, might be "insert a into b" or "load pallet with a". Execution level primitive tasks are associated with logical devices and are provided as input to the device drivers at the device dependent layer of the

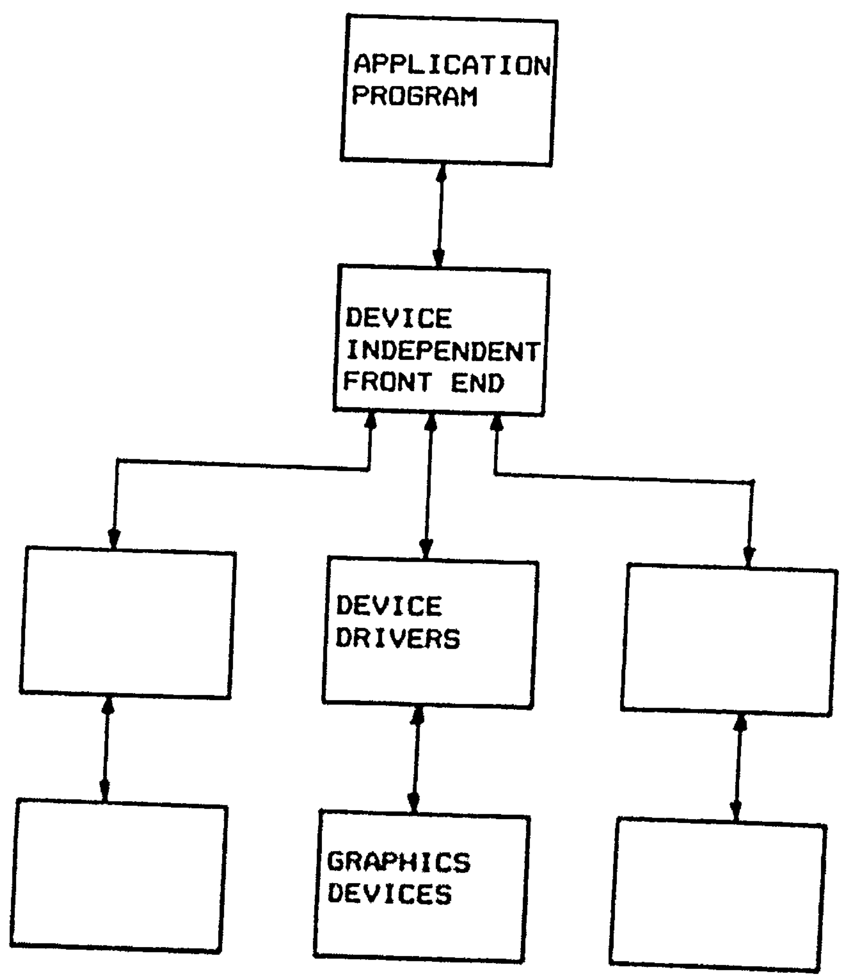


Fig. 7-8: Parts Of A Graphics System

control structure.

The set-up module coordinates the cell's devices and workstations during the part change-over process. This may include down-loading part programs or issuing commands which direct the device or workstation to reconfigure itself. For example, a logical robot may be instructed to change its gripper, or perhaps a logical robot and a logical transfer device are coordinated while the robot systematically reconfigures each pallet in the transfer system.

The run-time device coordination module integrates and coordinates the cell's devices and workstations during normal operation. This module exerts real-time control over the cell's devices and work stations through execution level primitive tasks which are associated with logical devices. These execution level primitive tasks are provided as input to the device dependent layer of the control structure.

The run-time device coordination module is capable of initiating and suspending device or workstation operation. It also provides multi-device synchronization when necessary. The run-time device coordination module is also capable of controlling the parallel execution of tasks on different cell devices. This point is important in that many approaches to cell or multi-device control are implemented in procedural programming languages. Without careful design such systems force sequential execution of tasks that should be executed in parallel. In a cell containing multiple devices, this may lead to poor equipment utilization. This potential problem is particularly pertinent to machining cells. Bourne and Fussell illustrate this weakness and suggest that control of manufacturing cells be based on a rule based programming language [31].

Control at this level is task oriented. The run-time coordination module provides the coordination and integration necessary for operation of multiple devices as a single unit, the cell. The run-time module is not concerned with the real-time internal device control. For example, the run-time control module might issue a command to a logical robot to suspend program execution and move to the "home" position. But during the move the run-time control module would not be concerned with the servo control at each of the robot's joints.

The sensor control module coordinates the various sensors which make up the cell's sensory system. This module controls the sensors and collects the raw sensory data required to satisfy the commands generated at the cell management layer. For example, the management layer may issue commands directing the execution layer to count the number of good assemblies produced in a given time period. The sensor control module utilizes the appropriate sensory system devices to collect the requested data. Execution level primitive (sensory) tasks, which are associated with logical sensory devices, are provided to the device dependent layer as input.

7.5.4. DEVICE DEPENDENT LAYER

The device dependent layer consists of device drivers. The device drivers process the commands (execution level primitives) provided by the execution layer and generate the device specific commands required to control the actual physical devices. Device drivers will vary both in implementation and in complexity. The device driver for a sophisticated robot may exercise control through communication with the robot controller, or it may be possible to implement the logical device driver within the robot's controller itself. Whereas the device driver for an asynchronous conveyor transfer system or simple pick and place robot might be quite different.

Sensors are also devices which are controlled at the device dependent layer. A sensor driver exists for each type of sensor that is interfaced to the control system. A driver is designed to meet specific requirements of the sensor, such as transmission speed and type of data encoding[26].

Sophisticated robots which are integrated with vision systems, or other sensory systems, at the device level, are treated as a single complex multi-capability device. As a result, such a device would have a single device driver at the device dependent layer of the control structure.

7.5.5. UTILITY MODULES & CELL DATA

A number of utility modules are defined which serve each of the four layers of the control structure. These modules act as front ends to the cell database and provide easy access to cell data. The utility modules are (1) program management, (2) material management, (3) tooling management, and (4) production status/data management. These modules service requests from the functional modules by accessing the cell database through the database management system, see figure 7-9.

The program management module is the mechanism used to manage the part programs for the programmable devices in the cell. This module provides for the management of the part program library maintained in the cell's database. It facilitates part program configuration management by enforcing change control procedures. It provides a mechanism for controlling the "part to part program" and "part program to device" associativity. The program management module is also used along with other modules in various layers of the control structure to do on-line device programming when necessary.

The materials management module provides for the management of the materials which are within cell control. It provides access to and maintains material status information for the cell. When component parts are delivered to the cell, the material management module is used to record their arrival and manages their storage within the cell.

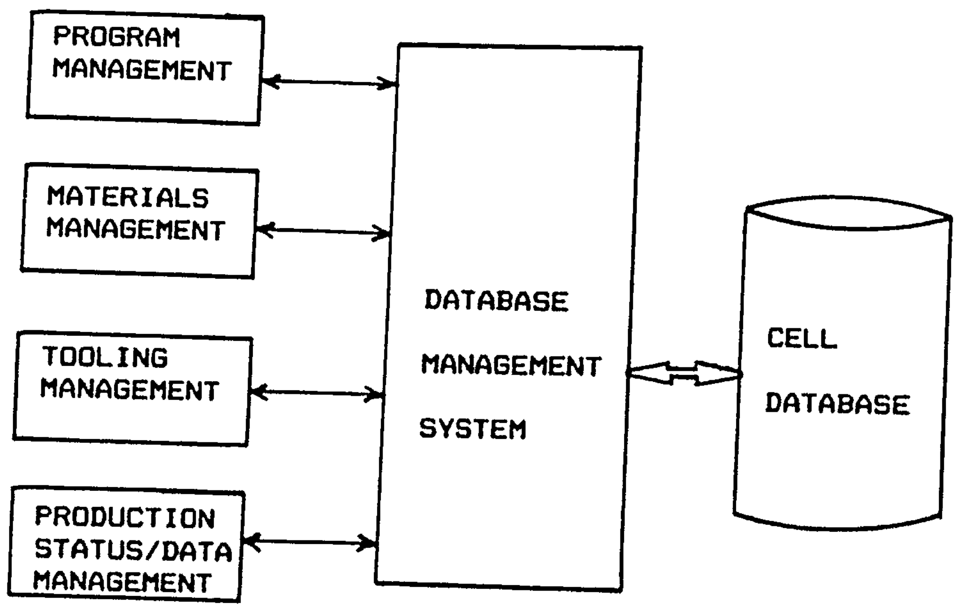


Fig. 7-9: Utility Modules & The Cell Database

The tooling management system is used to manage the tooling and fixtures used within the cell. Like the program management module it also provides a mechanism for controlling the "tool or fixture to part" and "tool or fixture to device" associativity. It is also used for tool or fixture change control and maintains the status of the cell's tooling and fixtures at all times.

The production status/data module manages production data and maintains cell status information. This module interfaces with the various layers of the control structure to collect, consolidate, abstract and interpret cell status and production data. On the other hand, this module also provides any requested cell status or production data to the functional modules which make up the control structure.

Each of these utility modules as well as any other functional module with a need for data, accesses the cell database through the cell's database management system. The cell's database management system satisfies all requests for data and provides the only method of inputting or extracting data from the database. The cell's database contains all the data required to operate the cell. The implementation of the cell's database is unimportant. What is important, is that there exist the logical equivalent of a cell database, where the data required to efficiently operate the cell is maintained. Once computer integrated manufacturing (CIM) becomes a reality throughout the manufacturing operation, a logical equivalent of a cell level database may very well be implemented on other company databases which utilize the same data, thereby eliminating the need for data redundancy.

7.6. SUMMARY

This chapter began by stating that the objective was to develop a cell level control structure that was easily maintained and modified, and could be applied across a whole class of assembly (manufacturing) cells. Some important differences between automated assembly cells and machining cells were demonstrated, and their impact on the control sys'em was discussed. The need for cell control in automated assembly cells was said to be the result of the need to coordinate the interactions of multiple devices in order to achieve real integration, and cell control was identified as the "integrating agent" that took the final step in making a collection of devices into a system for assembly. The hierarchical control approach being developed at the National Bureau of Standards was discussed. This control rationale is based on task decomposition, modularization of control, and the use of standard interfaces. A layered control structure was developed in the second part of this chapter. This layered approach is based upon many of the same principles applied in the NBS hierarchical control system. However, the scope of this control structure is limited to cell control. The control structure consists of four control layers, a set of utility modules, and a cell database, see figure 7-10.

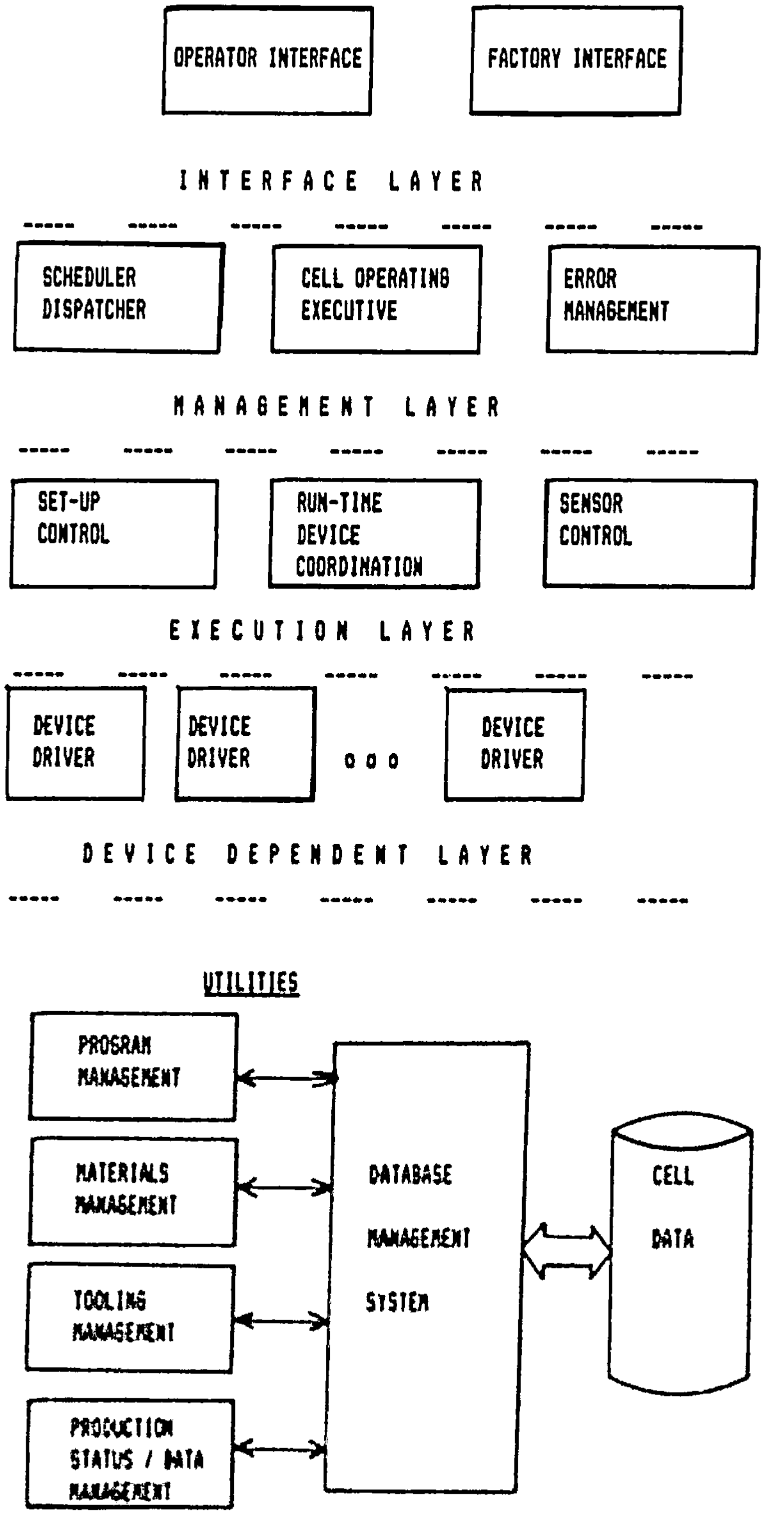


Fig. 7-10: Complete Layered Control Structure

8. RECOMMENDATIONS FOR FURTHER STUDY

As the economic considerations begin to make flexible manufacturing cells for assembly more appealing, there will be a rush to begin to interconnect robots, material handling systems, sensor systems, and various other devices in order to form assembly cells. But without careful design and further development in cell integration techniques, these new efforts will fall short of their intended goals. Although interconnection may not be difficult, achieving effective integration while maintaining flexibility and adaptability is much more elusive.

There are several areas in which progress must be made before off-the-shelf solutions to flexible automated assembly cells become available. One of these areas is manipulator programming languages. The flexibility of automated assembly systems containing programmable devices will be determined to a large extent by the ease with which these devices are programmed. There is a need for object level programming languages where the programmer describes moves in terms of the objects being manipulated.

Such languages utilize predefined object models and world models as a source of geometric and positional data, so that the programmer can symbolically manipulate objects and define movement.[32] Similarly there is a need to expand existing work on languages for describing assemblies and assembly processes.[10,33] The integration of such systems with CAD/CAM databases where certain geometrical and process data already exists, should also be pursued.

Task decomposition is another area requiring more work. The definition of tasks and the methodology used in the decomposition of tasks into subtasks should be formalized, and if possible customized to manufacturing cell control. The task decomposition methodology ultimately has a major impact on the control system's flexibility and capacity for expansion. The ideas of primitive tasks and task classes for different types of manufacturing, such as machining and assembly, need to be investigated and refined. Given that the whole concept of task decomposition is reasonable, it should be possible to decompose a high level goal into a set of standard tasks at each layer of the control structure, and finally arrive at a standard set of primitive tasks for the device dependent layer. This melds quite nicely with object level device programming. If a standard set of low level (relative to the cell) tasks are defined for the device level, device controllers and device programming languages can be built to implement these standard object level tasks.

Control system implementation is another area requiring further development. Although control structures and functional modules have been defined, their implementation has received considerably less attention. Bourne and Fussell have pointed out that:

> "A procedural language design, and a single processor computer architecture are not appropriate for the tasks of a cell management system."

Instead, Bourne and Fussell have opted to develop a rule based language which is constrained by a set of grammatical constructs where "the rules are executed as they are needed by the cell rather than being forced into a particular sequence."[31] The team developing the hierarchical control systems at the National Bureau of Standards have attempted to overcome the deficiencies of procedural language structures by the use of state tables to implement decision making.[18] Admittedly, most procedural languages in themselves do not provide for parallel execution or concurrency, and may not be as convenient as state tables for encoding complex decision logic. But, as is always the case, there are tradeoffs. State tables and rule based languages may not always be easy to use and easily modified. Clearly, control system implementation requires further development.

Collision avoidance and multiple device interference is an area which will be critical to the success of sophisticated assembly systems. As assembly systems become more adaptive and begin to make dynamic run-time adjustments, it will be critical that devices which share the same workspace with other devices, be able to predict and avoid collisions. Work in this area will have a significant impact on assembly systems.

9. REFERENCES

1. Black, J. T., "Cellular Manufacturing Systems Reduce Setup Time, Make Small Lot Production Economical" Industrial Engineering, Vol. 15, No. 11, (November, 1983)

2. Groover, M. P., Automation, Production Systems, and Computer-Aided Manufacturing, Prentice-Hall, Inc., 1980

3. DiPietro, F. A., "Organization of Assembly Cells by Robotics" Robots VIII Conference Proceedings, June 4-7, 1984

4. Dumolien, W. J., and Santen, W. P., "Cellular Manufacturing Becomes Philosophy of Management at Components Facility" Industrial Engineering, Vol. 15, No. 11, (November, 1983)

5. Dunlap, G. C., and Hirlinger, C. R., "Well Planned Coding, Classification System Offers Company-Wide Synergistic Benefits" Industrial Engineering, Vol. 15, No. 11, (November, 1983)

6. Soldner, K. T., and Spitznagel K. L., "A Multi-Task Robotic Workcell for Electronic Assembly" Robots VIII Conference Proceedings, June 4-7, 1984

7. Bortz, A. B., "Robots in Batch Manufacturing, the Westinghouse APAS Project" Robotics Age, Vol. 6, No. 1, (January 1984)

8. VanderBrug, G. J., "CYBERVISION: A System for Flexible Robotic Assembly" Electro/82 Conference Record, May 25-27, 1982

9. Hinson, R., "Analyze Assembly Techniques in Applying Robots" Industrial Engineering, Vol. 15, No. 12, (December, 1983)

10. Rembold, U., and Blume, C., "Programming Languages and Systems for Assembly Robots" Computers In Mechanical Engineering, Vol. 2, No. 4, (January 1984)

11. Rooks, B., "Programmable Assembly Systems" Assembly Automation, International Quarterly, Vol. 3, No. 3, (August 1983)

12. Mutter, R. F., "Implementation Considerations in Process-Oriented Robotic Workcells" Robots VIII Conference Proceedings, June 4-7, 1984

13. Hoffman, B. D., Pollack, S. H., Weissman, B., "Vibratory Insertion Process: A New Approach to Non-Standard Component Insertion" Robots 8 Conference Proceedings, June 4-7, 1984

14. Fisher, R. P., Falkner, C. H., "The Complex Art of Balancing Robotic Assembly Lines" Robotics Today, Vol. 5, No. 6, (December 1983)

15. Cutkosky, M. R., Fussell, P. S., Milligan, R., Jr., "The Design of a Flexible Machining Cell for Small Batch Production" Journal of Manufacturing Systems, Vol. 3, No. 1, 1984

16. Albus, J. S., Barbera, A. J., Nagel, R. N., "Theory and Practice of Hierarchical Control" Twenty-third IEEE Computer Society International Conference, September 1981

17. Barbera, A. J., Fitzgerald, M. L., Albus, J. S., Haynes, L. S., "RCS: The NBS Real-Time Control System" National Bureau of Standards, Industrial Systems Division

18. Haynes, L. S, Barbera, A. J., Albus, J. S., Fitzgerald, M. L., McCain, H. G., "An Application Example of the NBS Robot Control System" Robotics & Computer-Integrated Manufacturing, Vol. 1, No. 1, 1984

19. Albus, J. S., McLean, C. R., Barbera, A. J., Fitzgerald, M. L., "Hierarchical Control for Robots in an Automated Factory" 13th ISIR/Robots 7 Symposium Proceedings, Chicago, Illinois, April 1983

20. Albus, J. S., Barbera, A. J., Fitzgerald, M. L., "A Control System for an Automated Manufacturing Research Facility" Robots 8 Conference Proceedings, June 4-7, 1984

21. Fussell, P. S., Wright, P. K., Bourne, D., "A Design of a Controller as a Component of a Robotic Manufacturing System" Journal of Manufacturing Systems, Vol. 3, No. 1, 1984

22. Thomas, R., "Designing Controls for Robotic Work Cells" Industrial Engineering, May 1983

23. Wilkins, A., "SCAMP Comes in Modules" The FMS Magazine, Vol. 1, No. 1, (October 1982)

24. Graiser, H. N., "Automated Assembly Facility for Consumer Electronics Products" Robots 8 Conference Proceedings, June 4-7, 1984

25. Henderson, J. A., Hosier, R. N., "New Robotic Systems Change the Electronic Assembly Factory" Robots 8 Conference Proceedings, June 4-7, 1984

26. Demers, K. P., Walsh, P. M., "Sensor-based Real-time Robot Control Systems" Robotics Today, Vol. 5, No. 3, (June 1983)

27. Lyman, J., Madni, A. M., "Operator Roles in Robotics" Robotics Age, Vol. 6, No. 1, (January 1984)

28. Lee, M. H., Hardy, N. W., Barnes, D. P., "Error Recovery in Robot Applications" Proceedings of the Sixth British Robot Association Annual Conference, May 1983

29. Newman, W. M., Sproull, R. F., Principles of Interactive Computer Graphics, 2nd edition, Mcgraw-Hill, 1979

30. Foley, J. D., Van Dam, A., Fundamentals of Interactive Computer Graphics, Addison-Wesley, 1982

31. Bourne, D. A., Fussell, P., "Designing Languages for Programming Manufacturing Cells" Electro/82 Conference Record, May 25-27, 1982

32. Kempf, K. G., "Robot Command Languages and Artificial Intelligence" Robots 6 Conference Proceedings, March 2-4, 1982

33. Popplestone, R. J., Ambler, A. P., Bellos, I. M., "An Interpreter for a Language for Describing Assemblies" Artificial Intelligence, Vol. 14, 1980

10. VITA

Joseph A. Wetzel was born August 9, 1958 in Gettysburg, Pennsylvania, to Robert D. and Bettye L. Wetzel. He graduated Summa Cum Laude in December 1981 from Shippensburg State College, where he earned the Bachelor of Science in Computer Science. While attending Shippensburg State College he was elected to Kappa Mu Epsilon national honorary mathematics society. Prior to attending Lehigh University he was employed as a scientific programmer analyst working in the development of computer aided engineering software.